



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería Industrial

Unidad de Posgrado

**Desarrollo de un modelo basado en datos a partir de
señales de vibración para la detección de fallos en un
compresor reciprocante de simple efecto doble etapa**

TESIS

Para optar el Grado Académico de Doctor en Ingeniería Industrial

AUTOR

Adriana del Pilar GUAMÁN BUESTÁN

ASESOR

Juan Manuel CEVALLOS AMPUERO

Lima, Perú

2019



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Guamán, A. (2019). *Desarrollo de un modelo basado en datos a partir de señales de vibración para la detección de fallos en un compresor reciprocante de simple efecto doble etapa* [Tesis de doctorado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Industrial / Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.



**UNIVERSIDAD NACIONAL
MAYOR DE SAN MARCOS**

Universidad del Perú, DECANA DE AMÉRICA

UNIDAD DE POSGRADO

ACTA DE SUSTENTACIÓN N° 06-UPG-FII-2019

**SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO
DE DOCTORA EN INGENIERÍA INDUSTRIAL**

En la ciudad de Lima, del día veinte del mes de marzo del dos mil diecinueve, siendo las catorce horas, en acto público se instaló el Jurado Examinador para la Sustentación de la Tesis titulada: **"DESARROLLO DE UN MODELO BASADO EN DATOS A PARTIR DE SEÑALES DE VIBRACIÓN PARA LA DETECCIÓN DE FALLOS EN UN COMPRESOR RECIPROCANTE DE SIMPLE EFECTO DOBLE ETAPA"**, para optar el Grado Académico de Doctora en Ingeniería Industrial.

Luego de la exposición y absueltas las preguntas del Jurado Examinador se procedió a la calificación individual y secreta, habiendo sido Aprobada con la calificación de 18 (Muy bueno).

El Jurado recomienda que la Facultad acuerde el otorgamiento del Grado Académico de Doctora en Ingeniería Industrial, al **Mg. GUAMÁN BUESTÁN, ADRIANA DEL PILAR**.

En señal de conformidad, siendo las 15:45 horas se suscribe la presente acta en cuatro ejemplares, dándose por concluido el acto.


Dr. TINOCO GÓMEZ, OSCAR RAFAEL
Presidente


Dr. ALVAREZ MERINO, JOSÉ CARLOS
Miembro


Dr. CRUZ ORNETTA, VÍCTOR MANUEL
Miembro


Dr. WONG CABANILLAS, FRANCISCO JAVIER
Miembro


Dr. CEVALLOS AMPUERO, JUAN MANUEL
Asesor

INFORMACIÓN

Código ORCID del autor: <https://orcid.org/0000-0002-5184-6239>



Código ORCID del asesor: <https://orcid.org/0000-0001-8612-9128>

Grupo de investigación: Grupo de Investigación y Desarrollo en Tecnologías Industriales (GIDTEC)

Institución financiada parcial o totalmente la investigación: Universidad Politécnica Salesiana. Ecuador.

Ubicación geográfica donde se desarrolló la investigación: Laboratorio de vibraciones mecánicas de la Universidad Politécnica Salesiana, Cuenca, Ecuador (Calle Vieja 12-30 y Elia Liut)

Año o rango de años que la investigación abarcó: 2 años

DNI: 0301534582

AGRADECIMIENTO

Agradezco a Dios, dador de todo bien, principio y fin de todo cuanto he realizado en mi vida. A la Universidad Politécnica Salesiana por brindarme la oportunidad de cursar este programa doctoral y por el apoyo en este proyecto. A mis compañeros del Grupo de Investigación en Desarrollo y Tecnologías Industriales GIDTEC, Ph.D Diego Cabrera, Ph.D Vinicio Sánchez y Ph.D Mariela Cerrada por ser guías y amigos en el desarrollo de este proyecto. A mi tutor Ph.D Juan Cevallos por su acertado direccionamiento. A Cristian Torres y Franco Cajas por su contribución al presente proyecto. A Jean Carlos Macancela y Pablo Lucero por su apoyo y compañía en largas jornadas de trabajo. Finalmente agradezco a todos quienes contribuyeron para que mis estudios lleguen a un feliz término.

DEDICATORIA

A mis padres Alicia y Florencio y a mis hermanos Nancy, Glenda y David por ser la noche y el día en todos mis proyectos.

INDICE GENERAL

CAPÍTULO 1. INTRODUCCIÓN	1
1.1 Situación problemática.....	1
1.2 Formulación del problema	3
1.2.1 Problema general.....	4
1.2.2 Problemas específicos	4
1.3 Justificación práctica.....	4
1.4 Justificación teórica.....	5
1.5 Objetivos de la investigación	6
1.5.1 Objetivo general	6
1.5.2 Objetivos específicos	6
CAPÍTULO 2. MARCO TEÓRICO	8
2.1 Marco filosófico y epistemológico de la investigación.....	8
2.1.1 Marco filosófico	8
2.1.2 Marco epistemológico	12
2.2 Antecedentes de investigación	15
2.3 Bases teóricas	19
2.3.1 Compresores.....	19
2.3.2 Tipos de compresores.....	20
2.3.3 Fallos comunes en compresores reciprocantes	24
2.3.4 Mantenimiento de los compresores.....	27
2.3.5 Mantenimiento basado en la condición.....	27
2.3.6 Mantenimiento basado en la condición frente al mantenimiento centrado en la confiabilidad.	30
2.3.7 Monitoreo de la condición	31
2.3.8 Aprendizaje automático (Machine Learning)	36
2.3.9 Redes neuronales artificiales.....	43
2.3.10 Redes neuronales recurrentes	47
2.3.11 Optimización Bayesiana (OB)	53
2.3.12 Marcos conceptuales o glosario	54
CAPÍTULO 3. METODOLOGÍA	57
3.1 Tipo y diseño de la investigación.....	57

3.2	Unidad de análisis	58
3.3	Población de estudio	59
3.4	Tamaño de la muestra	60
3.5	Selección de la muestra	60
3.6	Técnicas de recolección de datos	61
3.7	Análisis de la información	61
3.8	Hipótesis.....	62
3.8.1	Hipótesis general.....	62
3.8.2	Hipótesis específica.....	62
3.9	Identificación de las variables	63
3.10	Operacionalización de las variables	64
3.11	Matriz de consistencia.....	64
CAPÍTULO 4. MODELO DE DETECCIÓN DE FALLOS PROPUESTO.....		67
4.1	Guía de adquisición de las señales de vibración	68
4.1.1	Descripción del banco de experimentación.....	68
4.1.2	Sistema de adquisición de las señales de vibración	69
4.1.3	Configuración del banco de experimentación.....	72
a.	Configuración del compresor	72
b.	Configuración de los parámetros del banco	75
c.	Configuración de las válvulas del compresor	75
4.1.4	Nomenclatura de los fallos en las válvulas del compresor	78
4.1.5	Base de datos.....	80
4.1.6	Software utilizado para la adquisición de datos.....	85
4.2	Pre-procesamiento.....	87
4.2.1	Ingreso de la señal.....	88
4.2.2	División del conjunto de señales.....	89
4.2.3	Ventaneo de la señal y extracción de las características	90
4.2.4	Estandarización de la señal	93
4.2.5	Incremento de las series de tiempo (señales)	93
4.2.6	Aplicación de software para el pre-procesamiento	94
4.2.7	Programación para el desarrollo de la metodología de pre-procesamiento.	97
4.3	Modelamiento con redes neuronales recurrentes LSTM y optimización bayesiana.	100

4.3.1 Construcción de la red neuronal con búsqueda bayesiana de los hiperparámetros.....	100
4.3.2 Aplicación de software para la creación del modelo	104
4.3.3 Programación para la creación del modelo	107
4.3.4 Software utilizado para la generalización del modelo.	108
4.3.5 Programación para la evaluación del modelo	110
CAPÍTULO 5. RESULTADOS Y DISCUSIÓN	112
5.1 Análisis e interpretación de resultados de la adquisición de datos y del pre-procesamiento.	112
5.1.1 Prueba de la hipótesis específica 1	116
5.2 Análisis e interpretación de resultados de la creación del modelo.....	118
5.2.1 Prueba de la hipótesis específica 2.....	120
5.3 Análisis e interpretación de resultados de la generalización del modelo.....	121
5.3.1 Prueba de la hipótesis específica 3	127
CONCLUSIONES.....	129
RECOMENDACIONES.....	132
REFERENCIAS BIBLIOGRÁFICAS	133
ANEXO A. Códigos de programación para el pre-procesamiento.....	147
ANEXO B. Códigos de programación para la creación del modelo.....	152
ANEXO C. Códigos de programación para evaluación del modelo.....	155

INDICE DE FIGURAS

Figura 1. Pasos del mantenimiento basado en la condición.....	2
Figura 2. División de la ciencia.	12
Figura 3. Estructura de un sistema de compresión.....	19
Figura 4. Tipos de compresores	20
Figura 5. Configuración de un sistema de compresión recíproco.....	21
Figura 6. Esquema de un funcionamiento de compresión de simple efecto	21
Figura 7. Diagrama P-V de un compresor reciprocante de dos etapas.	22
Figura 8. Ciclo ideal de compresión	22
Figura 9. Ciclo real de compresión	23
Figura 10. Esquema general de detección de fallos basado en modelo matemático..	29
Figura 11. Esquema de detección de fallos basado en datos.....	29
Figura 12. Principio básico de medición de la vibración	33
Figura 13. Aplicación de houldout para un conjunto de cuatro clases.....	38
Figura 14. Aprendizaje supervisado de dos clases.....	39
Figura 15. Clasificación por aprendizaje no supervisado	40
Figura 16. Estructura de una neurona biológica.....	44
Figura 17. Estructura de una neurona artificial con una entrada.....	45
Figura 18. Esquema de una red neuronal	46
Figura 19. Esquema de una red neuronal recurrente simple	47
Figura 20. Modelos de redes neuronales recurrentes	48
Figura 21. Descenso del gradiente	50
Figura 22. Estructura de una red LSTM.....	51
Figura 23. Compresor reciprocante de simple efecto doble etapa EBG250	59
Figura 24. Metodología para la construcción de modelos LSTM a partir de señales de vibración para el diagnóstico de fallos de un compresor	67
Figura 25. Banco de experimentación.....	68
Figura 26. Unidades del banco de experimentación	69
Figura 27. Esquema del sistema de adquisición de las señales de vibración.....	70
Figura 28. Ubicación del acelerómetro en el Compresor.....	72
Figura 29. Configuración del compresor EBG250	73
Figura 30. Esquema de funcionamiento del compresor EBG250	74

Figura 31. Vista en corte del compresor	76
Figura 32. Principio de funcionamiento de la válvula tipo placa de simple efecto. ..	76
Figura 33. Orientación de montaje de la válvula	78
Figura 34. Esquema 3D de la localización de las válvulas: a) vista superior y b) vista lateral en corte	78
Figura 35. Codificación para la identificación de válvulas.....	79
Figura 36. Vista superior del cabezote del compresor	79
Figura 37. Interfaz gráfica del programa de adquisición de señales.	85
Figura 38. Señal de vibración de un ciclo de giro del cigüeñal en el dominio del tiempo del compresor EBG250 adquirida por el acelerómetro A1	86
Figura 39. Señales de vibración correspondientes al desgaste del asiento de diferentes válvulas	87
Figura 40. Pre-procesamiento de las señales de vibración.....	88
Figura 41. Esquema de la base de datos.....	89
Figura 42. División del conjunto de señales adquiridas.....	90
Figura 43. Etapa de pre-procesamiento.....	91
Figura 44. Archivos que contienen las señales adquiridas.....	94
Figura 45. Señal a ser pre-procesada.....	95
Figura 46. Ventana deslizante y extracción de indicadores	96
Figura 47. Aplicación de una ventana de corte	96
Figura 48. Conjunto de señales estandarizadas.....	97
Figura 49. Diagrama general del pre-procesamiento	98
Figura 50. Función Parametric_Generator.....	99
Figura 51. Representación de la red LSTM	104
Figura 52. Datos de entrada a la red neuronal en su fase de entrenamiento	106
Figura 53. Exactitud del modelo generado	107
Figura 54. Resultado de la arquitectura neuronal LSTM.....	107
Figura 55. Optimización de hiperparámetros.....	108
Figura 56. Identificación de los cinco mejores modelos LSTM	109
Figura 57. Estadísticos para los cinco mejores modelos.....	109
Figura 58. Comparación para con otros enfoques.....	110
Figura 59. Programación para la evaluación del modelo	111
Figura 60. Cantidad de señales luego del pre-procesamiento	113

Figura 61. Comparación del error con respecto a los parámetros de pre-procesamiento	115
Figura 62. Metodología para la construcción de modelos LSTM a partir de señales de vibración para el diagnóstico de fallas de un compresor	119

ÍNDICE DE CUADROS

Cuadro 1. Fallos comunes en compresores reciprocantes.....	25
Cuadro 2. Clasificación de fallos en rodamientos según ISO 15243:2004.....	26
Cuadro 3. Representación del conjunto de datos D	38
Cuadro 4. Matriz de confusión para tres clases.....	41
Cuadro 5. Identificación de las variables	63
Cuadro 6. Operacionalización de las variables	65
Cuadro 7. Matriz de consistencia	66
Cuadro 8. Características del acelerómetro.....	70
Cuadro 9. Características de la tarjeta NI cDAQ-9234.....	71
Cuadro 10. Características del chasis NI cDAQ-9188.....	71
Cuadro 11. Datos técnicos del compresor modelo EBG250.....	73
Cuadro 12. Descripción general de las partes principales de la válvula tipo placa. ..	77
Cuadro 13. Localización de los fallos en las válvulas del compresor.....	80
Cuadro 14. Detalles de la construcción de los fallos y su localización.....	81
Cuadro 15. Configuración de fallos.	84
Cuadro 16. Características de la señal de vibración de un ciclo de giro de cigüeñal en el dominio del tiempo del compresor EBG250, adquirida por el acelerómetro.....	86
Cuadro 17. Distribuciones y límites que definen el espacio de búsqueda del hiperparámetro.	102
Cuadro 18. Intervalos para los parámetros de pre-procesamiento	113
Cuadro 19. Conjunto de modelos seleccionados para comparación de hiperparámetros.....	116
Cuadro 20. Comparación de modelos con y sin pre-procesamiento.....	117
Cuadro 21. Anova de un factor para los tres grupos.	117
Cuadro 22. Prueba Kolmogorov-Smirnov y métricas estadísticas de precisión para cada modelo LSTM.....	118
Cuadro 23. P-Valor de la prueba T entre pares de modelos.....	120
Cuadro 24. Prueba T para el par de modelos con y sin optimización de hiperparámetros.....	121
Cuadro 25. Comparación entre la propuesta y otros enfoques.....	122
Cuadro 26. Matriz de confusión del modelo LSTM-5 para el análisis de métricas.	123

Cuadro 27. Métricas para medir la eficiencia del modelo	123
Cuadro 28. Prueba Anova para comprobación de hipótesis tres.....	128

RESUMEN

Los compresores recíprocos son máquinas altamente utilizadas en las industrias por ser la principal fuente de aire comprimido. La aplicación de una estrategia de mantenimiento inadecuada para la detección temprana de fallos conduce al incremento de paros inesperados, incluso puede desencadenar eventos catastróficos para los procesos productivos. La detección de fallos en este tipo de máquinas resulta en la mayoría de casos complejo, por la dificultad para monitorear en tiempo real. En los últimos años se ha incrementado el uso de técnicas de modelamiento basado en datos para el diagnóstico de fallos. Estas técnicas requieren de grandes cantidades de datos que no siempre se pueden obtener pues generan altos costos y tiempo excesivo, que son difíciles de solventar desde el punto de vista económico y técnico. El presente trabajo se enfoca en tres aspectos: (i) la adquisición de datos, (ii) desarrollo de un método para el pre-procesamiento de las señales de vibración y (iii) propuesta de una metodología para el modelado basado en redes neuronales recurrentes *Long Short Term Memory* (LSTM) para el diagnóstico de fallos. El documento inicia con una descripción teórica de los fallos más comunes en compresores reciprocantes y se aborda el modelado basado en redes neuronales. Luego se realiza la experimentación para la adquisición de señales de vibración, donde se describe un plan experimental y los instrumentos utilizados. A continuación, se realiza el pre-procesamiento de las señales adquiridas, con la propuesta de un método para incrementar el número de series temporales informativas de la máquina y para obtener señales con menor resolución. Finalmente, se procede a la implementación y evaluación del modelo de diagnóstico, donde se entrena la red LSTM a partir de los mejores conjuntos de hiperparámetros obtenidos con un enfoque bayesiano que delimita el espacio de búsqueda en cada iteración. Este enfoque se aplica para detección de fallos en las válvulas de ingreso y de descarga del compresor. Al culminar la investigación, los resultados muestran que la exactitud del mejor modelo alcanza el 93%, seleccionado entre un grupo de mejores candidatos. Este resultado se validó comparándolo con tres enfoques clásicos y se evidencia claramente un mejor rendimiento para el modelado con redes LSTM.

SUMMARY

Reciprocating compressors are highly used machines in the industries because they are the main source of compressed air. The application of an inadequate maintenance strategy for early detection of failures leads to the increase of unexpected stoppages, even can trigger catastrophic events for the productive processes. The detection of failures in these type of machines results in the majority of complex cases because of the difficulty to monitor in real time. In recent years, the use of data-based modeling techniques for fault diagnosis has increased. These techniques require large amounts of data that cannot always be obtained because they generate high costs and excessive time, which are difficult to solve economically and technically. This paper focuses on three aspects: (i) data acquisition, (ii) development of a method for pre-processing vibration signals and (iii) a methodology proposal for modeling based on recurrent Long Short Term Memory (LSTM) neural networks for troubleshooting. The document begins with a theoretical description of the most common faults in reciprocating compressors and deals with the modeling based on neural networks. Experimentation is then carried out for the acquisition of vibration signals, which describes an experimental plan and the instruments used. Then, the pre-processing of the acquired signals is carried out, with the proposal of a method to increase the number of time series informative of the machine and to obtain signals with lower resolution. Finally, we proceed to the implementation and evaluation of the diagnostic model, which trains the LSTM network from the best sets of hyperparameters obtained with a Bayesian approach that delimits the search space in each iteration. This approach is applied to detect faults in the intake and discharge valves of the compressor. At the end of the research, the results show that the accuracy of failure recognition of the best model reaches 93%, selected among a group of best candidates. This result was validated by comparing three classic approaches and clearly demonstrates a better performance of modeling with LSTM networks.

CAPÍTULO 1. INTRODUCCIÓN

1.1 Situación problemática

La operación segura de los compresores es complicada sin un monitoreo de condición, además de la imposibilidad de realizar acciones de monitoreo sin detener el sistema (Guerra, 2013). Para este autor los fallos que se presentan en los compresores reciprocantes se distribuyen de la siguiente manera: 36% en válvulas, empaques 17.6%, rodamientos 8.8%, anillos del pistón 7.1%, bandas 6.8%, depósitos 6.8%, sistema de lubricación de cilindro 5.1%, instrumentación 5.1% y otros 3.4%. La falla ocasionada por las válvulas incide en el 50% del costo total de reparación.

Ante esta premisa N. K. Verma, Sevakula, Dixit, & Salour, (2016) sostienen que el Mantenimiento Basado en la Condición (MBC) es una estrategia adecuada para monitorear la condición de los compresores reciprocantes, puesto que permite la detección de fallos, a través de la identificación de cambios específicos en los parámetros como vibración, emisión acústica, temperatura, presión, entre otros. Es preciso señalar que la estrategia del MBC se da en tres etapas: (i) adquisición de datos, (ii) procesamiento de datos y (iii) toma de decisiones (Andrew K. S. Jardine, Lin, & Banjevic, 2006) como se representa en la Figura 1. En la adquisición de datos se obtienen y almacenan las señales de monitoreo de la condición; en el procesamiento de datos se trata y analiza la información obtenida y en la etapa de toma de decisiones se deciden las acciones basadas en los resultados de diagnóstico y pronóstico de fallos (Ahmad & Kamaruddin, 2012).

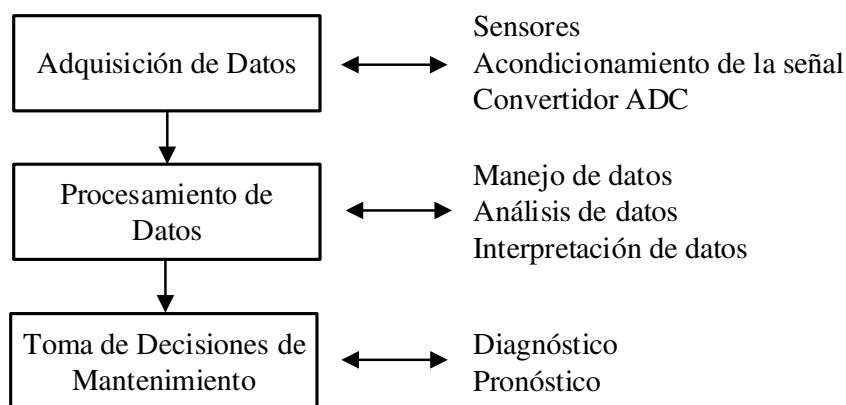


Figura 1. **Pasos del mantenimiento basado en la condición.** Tomado de (Deepam Goyal, Pabla, Dhami, & Lachhwani, 2017).

Como se puede apreciar en la Figura 1, la toma de decisiones de mantenimiento abarca dos aspectos importantes: el diagnóstico y el pronóstico. El diagnóstico se orienta a la detección, aislamiento e identificación de fallos cuando ya se han producido; no así el pronóstico, que se ocupa de la predicción en el tiempo de fallos (Deepam Goyal, Pabla, Dhami, & Lachhwani, 2017). En este caso es necesario considerar que la detección indica si hay anomalía en el comportamiento del sistema monitoreado, el aislamiento localiza el componente defectuoso y la identificación determina el tipo de fallo (Suzan Alaswad & Xiang, 2017). Es preciso señalar que la presente investigación se enfocará en la fase de diagnóstico, específicamente en la fase de detección de fallos.

Con respecto a lo expuesto en el párrafo anterior, es oportuno señalar que el diagnóstico puede aplicar diferentes enfoques, uno de ellos es el enfoque fundamentado en modelos que pueden ser agrupados en dos clases: modelos matemáticos y modelos basados en datos conocidos en inglés como *Data-Driven models*. Los primeros utilizan relaciones matemáticas entre algunas variables medibles para extraer información sobre posibles cambios causados por fallos. Pueden ser precisos y efectivos, sin embargo para ciertos sistemas no son factibles por la dificultad de enlazar las variables y por la complejidad para tomar mediciones de éstas (Isermann, 2006). A diferencia de éste, el modelo basado en datos surge como una

alternativa, se basa en la utilización de métodos de reconocimiento de patrones para un mapeo de las características en condiciones normales y en condiciones de fallo del sistema (Andrew K. S. Jardine et al., 2006), (Yin, Ding, Xie, & Luo, 2014).

Se han efectuado varias investigaciones sobre modelos de detección de fallos basados en datos en diferentes sistemas mecánicos como engranajes (Howard, Jia, & Wang, 2001), (Wang, 2002), rodamientos (Baillie & Mathew, 1997), (Loparo, Adams, Lin, Abdel-Magied, & Afshari, 2000), rotores (Oppenheimer & Loparo, 2002), (Sekhar, 2004) y herramientas de corte (Choi & Choi, 1996). En referencia a compresores reciprocantes se ha desarrollado clasificadores de fallos basados en señales de vibración, utilizando algoritmos genéticos (M. Ahmed, Smith, Gu, & Ball, 2014) y técnicas de redes neuronales artificiales (R. Ahmed, Sayed, Gadsden, Tjong, & Habibi, 2015), además de modelos matemáticos (Manepatil & Tiwari, 2006) y también se han planteado criterios para la evaluación de modelos (Kothamasu, Shi, Huang, & Leep, 2004). Como se puede evidenciar, el campo de los compresores es de mucho interés para el sector industrial e investigativo, existen significativos aportes en esta área, sin embargo hay mucho camino por recorrer para establecer un modelo con una metodología basada en datos a partir de señales de vibración, que facilite el diagnóstico de la condición de los compresores sin necesidad de intervenir directamente en la máquina. La presente investigación incursiona en este tema, específicamente en la detección de fallos para compresores reciprocantes de simple efecto doble etapa.

1.2 Formulación del problema

Para la resolución del problema de investigación se ha propuesto un problema general y tres problemas específicos que están formulados en forma de pregunta y se detallan a continuación.

1.2.1 Problema general

¿Es factible desarrollar un modelo basado en datos a partir de señales de vibración para detección de fallos en un compresor reciprocante de simple efecto doble etapa?

1.2.2 Problemas específicos

- a. ¿Cómo influye el pre-procesamiento de las señales adquiridas mediante la guía de adquisición de señales de vibración, en la construcción de un modelo basado en datos para la detección de fallos en compresores reciprocantes de simple efecto doble etapa?
- b. ¿Cómo se construye un modelo basado en datos a partir de la optimización de hiperparámetros para la detección de fallos en compresores reciprocantes de simple efecto doble etapa?
- c. ¿La evaluación del modelo generado a partir de señales de vibración, generaliza la detección de fallos en un compresor reciprocante de simple efecto doble etapa?

1.3 Justificación práctica

La globalización y el escenario competitivo en el sector industrial, demanda calidad en los procesos productivos, mayor rendimiento de los sistemas y alta eficiencia (Yin et al., 2014). Por otra parte, el creciente desarrollo de la tecnología ha conducido a la oferta de productos cada vez más complejos, lo que exige una mejor calidad y confiabilidad de los equipos. Esto ha llevado a determinar que el mantenimiento preventivo, que es un factor común en la mayoría de empresas, se vuelva cada vez más costoso, representando un gasto significativo para muchas de

ellas (Andrew K. S. Jardine et al., 2006). El MBC se presenta como la mejor alternativa para la planeación de sistemas de mantenimiento, sus beneficios radican en mejores procedimientos de mantenimiento y programación, mayor disponibilidad operativa de las máquinas y mejor calidad en los productos (James Li & Yu, 1995).

Una de las máquinas más utilizadas en la industria son los compresores reciprocantes, debido a su capacidad para comprimir casi cualquier mezcla de gases a altas presiones (Stiaccini, Galoppi, Ferrari, & Ferrara, 2016). Dada su incidencia en los procesos productivos, es importante garantizar la fiabilidad y disponibilidad para mantenerlos operando en condiciones adecuadas. De esta forma, el presente trabajo se orienta hacia la detección de fallos en compresores reciprocantes de simple efecto doble etapa, utilizando un modelo basado en datos a partir de las señales de vibración, pues se ha demostrado que estas son altamente efectivas en el monitoreo de la condición y diagnóstico de fallos en diferentes sistemas mecánicos (Li, Sanchez, Zurita, Cerrada, et al., 2016), (Zhao, Wang, Xing, & Gao, 2015). De acuerdo a lo expuesto, el presente trabajo es de gran relevancia en el ámbito industrial, pues aporta con datos que podrían servir a los gestores del mantenimiento para que consideren la estrategia del mantenimiento basado en la condición (MBC) como una oportunidad de mejora en sus procesos productivos.

1.4 Justificación teórica

La evolución del mantenimiento desde varias décadas, dio como inicio al mantenimiento correctivo, luego al mantenimiento preventivo y finalmente al mantenimiento predictivo. Al respecto se han dado una gran cantidad de aportes que le han posicionado como un factor clave dentro de los procesos productivos y ser visto como una oportunidad de mejora por la reducción de costos por paradas no programadas, seguridad para el operario y mejora en la calidad de los productos. En tal virtud, la tendencia actual se enmarca en estos principios y los esfuerzos en el ámbito investigativo para la detección de fallos en compresores, como los realizados

por M. Ahmed, Baqqar, Gu, & Ball, (2012) y Guerra & Kolodziej, (2014) han dado como resultado el avance de diversas estrategias para el monitoreo de la condición. Esto motiva a desarrollar una metodología orientada a compresores reciprocantes para la detección de fallos basados en datos de vibración que es un campo relativamente nuevo y con mucho interés, considerando que estas máquinas inciden directamente en el desarrollo normal de los procesos productivos. De esta manera, la presente investigación se propone contribuir con datos y referenciar una metodología que sirva como base para futuras investigaciones en el campo del monitoreo de la condición, basado en señales de vibración para su aplicación en compresores reciprocantes.

1.5 Objetivos de la investigación

Para resolver el problema planteado en la presente investigación se ha propuesto un objetivo general y tres objetivos específicos que se detallan a continuación.

1.5.1 Objetivo general

Desarrollar un modelo basado en datos de señales de vibración para la detección de fallos en compresores reciprocantes de simple efecto doble etapa.

1.5.2 Objetivos específicos

- a. Determinar cómo influye la aplicación de una técnica de pre-procesamiento de las señales adquiridas en la construcción de un modelo basado en datos para la detección de fallos en un compresor reciprocante de simple efecto doble etapa.

- b. Diseñar un modelo basado en datos a partir de la optimización de hiperparámetros para la detección de fallos en compresores reciprocantes de simple efecto doble etapa.
- c. Evaluar la capacidad de generalización del modelo basado en datos a partir de la señal de vibración para la detección de fallos en compresores reciprocantes de simple efecto doble etapa.

CAPÍTULO 2. MARCO TEÓRICO

Esta sección está constituida por el marco filosófico, los antecedentes de la investigación, las bases teóricas y el glosario de términos.

2.1 Marco filosófico y epistemológico de la investigación

2.1.1 Marco filosófico

El origen del método científico y la investigación tiene sus inicios en el siglo IV a.C, donde resaltan filósofos como Sócrates, Aristóteles y Platón que sugieren la necesidad de establecer un método para conseguir un fin a través de normas o reglas. La principal contribución de Sócrates (470 a.C. - 399 a.C) es su modo dialéctico de indagar, con esto pretendía alcanzar una comprensión neutral de los conceptos morales como el bien, la justicia y el amor, pues definió al alma como una combinación entre inteligencia y carácter (Gasparotti, 1996). Por su parte Platón (427 a.C-347 a.C) como buen discípulo de Sócrates trata temas filosóficos y políticos donde desarrolla la doctrina de las ideas, conjugando el pensamiento y la expresión para alcanzar una pluralidad, fundamentada en la asunción de la existencia de un mundo inteligible correspondiente a las ideas que se encuentra más allá de los objetos (Platón, 1968). De esta manera Platón orienta a la comprensión de la existencia de la materia-espíritu,

cuerpo-alma como elementos completamente separados. Sus ideas tuvieron gran influencia en Aristóteles (384-322 a.C) al considerar la experiencia para conocer la realidad; no obstante Aristóteles se diferenció porque rechazó la teoría de las ideas, sostuvo que existe relación entre el sujeto y el objeto, es así que en su obra “Ética a Nicómaco” relaciona la inteligencia y el carácter con la felicidad. Para Él, la virtud depende del contexto y de las diferentes situaciones que experimenta el individuo. Con esto Aristóteles sienta las bases de teoría del conocimiento para el desarrollo de la ciencia a través de la organización, recolección y clasificación de información del objeto de estudio, que en su caso fueron las ciencias naturales (Benavides, 2013).

Posterior a esta época surge otro protagonista del progreso de la ciencia, Galileo Galilei (1564- 1642), desarrolló procesos experimentales con el fin de analizar un fenómeno y monitorear sus variaciones, así Galileo trabaja en la construcción de un nuevo método científico basado en las matemáticas y la experimentación para la investigación de los fenómenos de la naturaleza como el análisis del movimiento de los cuerpos, que supondría ser la piedra angular para el desarrollo de la física en las siguientes generaciones (Márquez, 2017).

A finales del siglo XVI y a inicios del siglo XVII aparecen Bacon (1561-1626) y René Descartes (1596-1650), considerados los fundadores del pensamiento moderno. El principal aporte de Bacon es el método científico, pues remplace la lógica demostrativa y experimental a una experimental e inductiva, donde establece que para la experimentación debe haber una organización racional y metódica, con varias repeticiones a fin de poder generalizar los hechos. Su teoría empírica fue publicada en su documento *Novum Organum*, al sugerir la necesidad de efectuar un ciclo repetitivo de observación, hipótesis, experimentación y verificación; convirtiéndose así en uno de los pioneros del pensamiento científico moderno (Pavía & Soto, 2006). En forma paralela, Descartes fundamentó la corriente filosófica del racionalismo, según el cual la razón posibilita alcanzar la verdad, de forma que en su publicación “Discurso del Método” propuso descomponer problemas complejos en partes sencillas, haciéndolas más evidentes hasta poder explicar a través de la síntesis lo inicialmente considerado complejo (método deductivo). También su filosofía cartesiana de un mundo mecanicista tuvo gran influencia en la física clásica al publicar los “Principios Matemáticos de la Filosofía Actual” (Márquez, 2017).

Enmarcado en esta filosofía, Isaac Newton (1642-1727) perfecciona la construcción del método científico, su aporte se orientó al planteamiento de hipótesis cuyo fin era explicar las propiedades de un objeto de estudio. De esta manera su principal contribución se da en el campo de la física clásica y la física newtoniana, estableciendo las tres leyes del movimiento: (i) la ley del equilibrio, (ii) la ley de la dinámica y (iii) la ley de la acción y reacción, consolidándose como la base del desarrollo de todas las ciencias, especialmente de la mecánica (Ron, 2011). A partir del origen de la física clásica, se dan varias contribuciones, sin embargo entre ellas resalta Kant (1724-1804), sus aportes a la filosofía y la ciencia se basan en la crítica a la razón pura, la crítica a la razón práctica y la crítica del juicio al considerar que todo conocimiento inicia con la experiencia, instituyendo de esta forma las bases y condiciones para la generación de conocimiento científico. Su filosofía crítica diferencia los conceptos de certeza y creencia, orientando así al establecimiento de la experiencia y la racionalidad lógica como elemento fundamental para la certeza a modo de afirmación objetiva proveniente de la razón (Pavía & Soto, 2006).

Un siglo más tarde surge otro filósofo muy importante para el avance de la ciencia, Karl Popper (1902-1994), su principal contribución fue el principio de “Falsación”, mediante el cual sostuvo que el control empírico de la ciencia se efectúa con la posibilidad de falsar las hipótesis. Sostuvo que deben brindar la posibilidad de medir los resultados de los enunciados observables, de tal manera que si estos no se pueden verificar, la hipótesis no aporta a la verdad científica. Así Popper introduce la teoría del “Método Deductivo de Contrastar” y critica la aplicación de todo principio de inducción como mecanismo para generalizar las observaciones, pues sostiene que se conduciría a incoherencias lógicas. Con esto se plantea la necesidad de una explicación causal de un acontecimiento y propone dos clases de enunciados que podrían utilizarse como hipótesis: (i) enunciados universales y (ii) enunciados singulares, estos últimos se aplican de manera específica a un determinado acontecimiento y les llama *condiciones iniciales*, cuya finalidad es demostrar la causa de un fenómeno y a su vez señala que el efecto viene determinado por la predicción dado por el primer enunciado. (Popper, 2008).

Por otro lado Thomas Kuhn (2011) en su libro “La Estructura de la Revolución Científica” introduce el término *paradigma* donde lo define como “una visión

compartida sobre la ciencia y la realidad cuya investigación se basa en paradigmas compartidos que están sujetos a las mismas reglas y normas para la práctica científica”. Para este filósofo la investigación se da en una comunidad científica y no individualmente, así trata de explicar la validez de un hecho cuando ha sido asumido por un colectivo sin ponerlo en duda, mientras no existan anomalías que lleven a la invalidación del mismo, donde será necesario otra forma de hacerlo. Con esto Kuhn hace comprensible la verdad histórica de la naturaleza del conocimiento científico (Mayoral, 2017).

Finalmente Mario Bunge, (2014) en su obra “La Ciencia, su Método y su Filosofía”, proporciona una visión formalista de la ciencia, al sostener que no toda investigación científica procura el conocimiento objetivo. Entendiendo lo señalado, la ciencia no es totalitaria, pues no da precisamente un conocimiento de la realidad, para explicar mejor divide a la ciencia en dos clases: *formales* (ideales) y *fácticos* (materiales), para Bunge la ciencia formal (lógica y matemática) no se ocupa de los hechos, por sí mismas no dan un conocimiento de la realidad, más bien su objetivo es establecer correspondencia entre objetos formales con procesos que pertenecen a cualquier nivel de la realidad. Contrario a esto, el conocimiento fáctico o llamado también empírico proviene de sucesos y procesos que se alcanzan con la observación y/o experimentación. De este modo no emplean variables lógicas y la racionalidad es necesaria pero no suficiente para sus enunciados (Bunge, 2014). En resumen, la experiencia es el factor preciso para aseverar que una hipótesis referente a un hecho es válida, sin embargo no está determinada a ser la única. Por lo tanto el conocimiento fáctico aunque es racional sólo da una probabilidad, para lo cual utiliza inferencias inductivas y probables.

A continuación en la Figura 2 se resumen la propuesta de Bunge sobre las ciencias formales y fácticas, donde se puede apreciar que se proporciona una pauta clara para concebir un enunciado como probablemente verdadero partiendo de la experiencia. Este autor propone el uso de datos empíricos que junto con la experimentación se alcanza la realidad de los hechos y para el efecto considera necesario partir de una hipótesis. Se podría decir que este autor es uno de los mejores protagonistas de la ciencia en la última era, pues sus aportes conducen significativamente al desarrollo de la investigación científica.

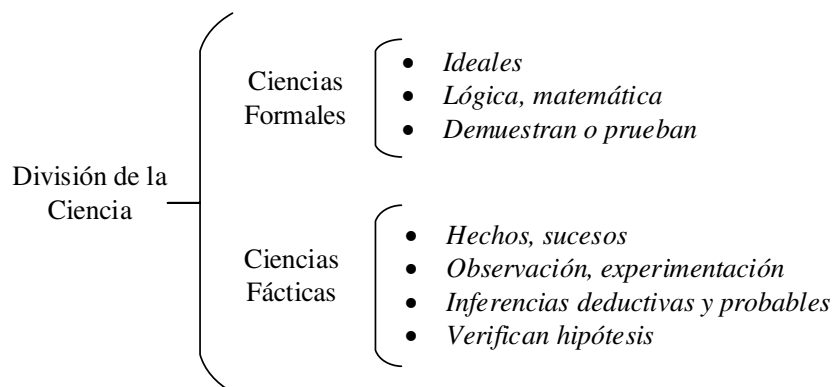


Figura 2. **División de la ciencia.** Elaboración propia en base a división de la ciencia efectuado por (Bunge, 2014).

Con el aporte de Bunge y de varios personajes a lo largo de los siglos, la evolución de la ciencia ha dado pasos significativos que han conducido al desarrollo de la ciencia en diferentes ámbitos. El tema de estudio de esta investigación no está exenta de esta evolución, pues el mantenimiento es un aspecto muy importante para el mejoramiento de los procesos productivos. Cada protagonista ha conducido directa o indirectamente al desarrollo de la detección de fallos, pues diversos estudios llevados a cabo datan la importancia y necesidad de mejorar las estrategias y técnicas de mantenimiento. A continuación se realiza un análisis retrospectivo de los aportes sobre la evolución de la detección de fallos en compresores.

2.1.2 Marco epistemológico

Desde los inicios de la humanidad, el hombre ha mantenido sus herramientas y equipos en condiciones de uso aceptable. Con la aparición de los primeros filósofos como Aristóteles, Platón y Sócrates, se planteó la necesidad de indagar sobre los fenómenos de la naturaleza, así la ciencia ha evolucionado, aportando consigo el desarrollo de muchas áreas, entre ellas el mantenimiento. El desarrollo visible del mantenimiento, data a finales del siglo XVIII y a inicios del siglo XIX. Con la revolución industrial, originada por la mecanización de los procesos productivos, surgen los pioneros del mantenimiento industrial, Frederick Winslow Taylor (1856-

1915), Henry Fayol (1841-1925) y Henry Ford (1863-1947). Taylor inicia con la organización de las tareas, permitiendo disminuir tiempos muertos y asignar un salario justo en función de la estandarización del trabajo. Por su parte Fayol se centró en el desarrollo de un modelo administrativo basado en tres aspectos: (i) la división del trabajo, (ii) la aplicación de un proceso administrativo y (iii) formulación de criterios técnicos como insumo para la función administrativa. Para consolidar las contribuciones realizadas, Fayol con su desarrollo de producción en masa, orientó a la especialización del trabajo como mecanismo para reducir costos de producción con la finalidad de obtener mayor beneficio económico (León, 1998). Los aportes generados por Taylor y Fayol, motivó la optimización de los sistemas mecánicos, de esta manera, en principio los operarios eran quienes realizaban reparaciones cuando los equipos y maquinaria dejaban de funcionar, así se da origen a la primera generación del mantenimiento con el *Mantenimiento Correctivo*. Otro aspecto importante de esta época es la aparición del término *falla* que se utilizó para determinar que las averías ocasionaban paros en la producción, obteniéndose los primeros registros estadísticos sobre las *tasas de falla* para la planificación de repuestos (Cardenal, 2018).

La segunda generación del mantenimiento se da durante la segunda guerra mundial, teniendo como referente a William Edwards Deming (1900-1993), conocido como el padre de la calidad. Sostuvo que “todo proceso presenta variabilidad y la medida en que ésta se reduzca existirá mejor calidad”. Tomando como referencia este principio, Walter Shewhart (1891-1967) introdujo las gráficas de control para diferenciar la variación entre causas aleatorias y causas especiales que permitiría predecir el comportamiento futuro de un proceso, esto supuso mejorar la productividad por la reducción de costos ocasionado por retrasos, reprocesos y fallas.

Las contribuciones dadas, cambió la concepción inicial del mantenimiento correctivo hacia el *Mantenimiento Preventivo* con el objetivo de mantener continuidad en la producción, pues en esta filosofía no sólo bastaba con conocer las fallas producidas en los equipos, también era necesario prevenirlas con acciones de carácter periódico que permita planificar tareas de mantenimiento sin paradas no programadas de la producción (León, 1998). Con las bases fundamentadas por Deming y Shewhart, se evidencia considerables reducciones en las tasas de falla y en forma paralela se aplican técnicas de prueba y ensayo que permitieron conocer el estado de las máquinas

a lo que se conoció como *Mantenimiento Predictivo*. Con esta filosofía fue posible detectar fallos a una edad temprana, para lo cual se emplearon dos orientaciones: (i) centrada en la confiabilidad y (ii) basada en la condición. De acuerdo a la primera, Greer (1960) propone la planificación del mantenimiento con el uso de estadísticas que indiquen la evolución histórica de las máquinas y McCall (1965) introduce la necesidad de optimización de las políticas de mantenimiento. Para la orientación del mantenimiento basado en la condición surgen los primeros estudios sobre la aplicación de técnicas de monitoreo y se desarrolla un primer estudio efectuado por Frankel y Kontorova (1938) para analizar la deformación plástica de los materiales utilizando la emisión acústica y más tarde Sohre (1968) propone el monitoreo de la condición utilizando señales de vibración. Esta técnica revolucionó el mantenimiento predictivo, pues sería el más utilizado en todos los campos de la industria hasta la actualidad (León, 1998; Rao, 1996), (Andrew K. S. Jardine et al., 2006).

Con la implementación del análisis de las señales de vibración y siendo la técnica más utilizada, han surgido múltiples aportes. Entre los trabajos más relevantes se encuentra el algoritmo propuesto por Cooley y Turkey para el análisis de la vibración y ruido; adicionalmente Cooley, Lewis y Welch trabajaron con la transformada de Fourier para el análisis del espectro. Más tarde en 1970 se desarrolló el método Kurtosis que fue de gran utilidad para el análisis vibracional de las máquinas y a partir de 1972 investigadores como J. Broderick, C. Osuagwu y R. Randall llevan el análisis de la vibración hacia diferentes elementos mecánicos como rodamientos, engranajes entre otros.

Posteriormente se da origen a la tercera generación, pues en las décadas de los ochenta y noventa se extiende el análisis de vibración hacia la aplicación industrial, además se introduce el uso de la computación para la adquisición de datos, el control y análisis de señal. Esto revolucionó nuevamente el tratamiento del mantenimiento predictivo hasta conducir a la aplicación de sistemas expertos en lenguaje, entre ellos las *Redes Neuronales*. Enmarcados en estos aportes y con la creciente tendencia del mantenimiento predictivo basado en la condición, la presente investigación utiliza las señales de vibración para la detección de fallos en compresores, aplicando las redes neuronales recurrentes.

2.2 Antecedentes de investigación

Grajales, Sánchez, & Pinzón, (2006) definen al mantenimiento como el conjunto de acciones destinadas a mantener o reacondicionar un componente, equipo o sistema, en un estado en el cual sus funciones pueden ser cumplidas en condiciones óptimas. Actualmente en la industria, el mantenimiento representa entre un 15% y 70% de los costos totales de producción (Bevilacqua & Braglia, 2000). El origen radica en la complejidad de la interacción de los sistemas a partir de componentes que con el tiempo pueden fallar de forma individual o secuencial (Diego Galar, Thaduri, Catelani, & Ciani, 2015). De esto surge la necesidad de seleccionar diferentes estrategias de mantenimiento que orienten a minimizar el peligro para el operario, reducir paradas no programadas y contribuir a una producción continua que repercutiría en una reducción de costos para los procesos productivos.

Las estrategias de mantenimiento se pueden clasificar en: mantenimiento correctivo, mantenimiento preventivo y mantenimiento predictivo (Coria, Maximov, Rivas-Dávalos, Melchor, & Guardado, 2015). El *mantenimiento correctivo* es la estrategia más antigua y se basa en intervenir en los sistemas cuando se produce una falla, por lo que no requiere una planificación previa. Sin embargo, genera altos costos por paradas no programadas en situaciones que pueden conducir a inconvenientes en el proceso productivo y más aún, representa un alto riesgo de seguridad para los operarios (Kothamasu, Huang, & VerDuin, 2006).

El *mantenimiento preventivo* se refiere a intervenciones periódicas que se llevan a cabo para reducir la posibilidad de fallas imprevistas (Deepam Goyal et al., 2017), surgió como una oportunidad para economizar costos operacionales (Grajales et al., 2006); no obstante, puede generar altos costos porque en algunos casos se ha asumido un reemplazo de elementos de manera preventiva en escalas de tiempo como: la edad, el periodo de funcionamiento, el número de uso y la experiencia del operario (Malik, 1979), (Chang, 2014). Con el avance de la tecnología surge el mantenimiento predictivo que se basa en el monitoreo de la condición (MC) del sistema para detectar fallos antes que originen una falla (Coria et al., 2015). Se refiere fallo a la desviación

no permitida de al menos una propiedad o característica del sistema en condición aceptable, usual o estándar; mientras que una falla es la interrupción permanente de la capacidad de un sistema para realizar una función requerida bajo condiciones de funcionamiento específicas (Isermann, 2006). En el campo industrial es recomendable detectar fallos en etapas tempranas o que sean aún incipientes para tomar acciones adecuadas antes que el sistema llegue a la falla.

El *mantenimiento predictivo* a su vez tiene dos enfoques, mantenimiento basado en la condición (MBC) y mantenimiento centrado en la confiabilidad (MCC) (Kothamasu et al., 2006). El MCC tiene su origen en la industria aeronáutica, cuyo objetivo es maximizar la confiabilidad del sistema y la reducción de costos por paradas inesperadas (Niu, Yang, & Pecht, 2010). Por otra parte, el MBC es una estrategia para medir diversos parámetros físicos de la máquina como vibración (Li, Valente de Oliveira, et al., 2016), emisión acústica (Li, Sanchez, Zurita, Cerrada Lozada, & Cabrera, 2016), entre otros; proporcionando así, un monitoreo regular de distintos parámetros de los componentes del sistema (Deepam Goyal & Pabla, 2015). Esta estrategia de mantenimiento recopila y evalúa información en tiempo real, permitiendo tomar decisiones basadas en información. Resultado de esto es la reducción de costos por la mejora de la fiabilidad del sistema (Suzan Alaswad & Xiang, 2017).

Los sistemas de aire comprimido, tienen gran incidencia en los procesos industriales pues, son considerados como el cuarto recurso más importante en las industrias después de la energía, el agua y el combustible (Guerra, C., 2013). El motivo que origina su mayor uso es su limpieza, disponibilidad, fácil uso y eficiencia energética que oscila entre el 90% y 95% (M. Yang, 2009). Adicionalmente, los compresores reciprocantes de simple efecto doble etapa son lo más utilizados por su capacidad para alcanzar altas tasas de presión que está en entre 10 a 15 bar. El aire obtenido de estas máquinas tiene una gran variedad de aplicaciones para la operación de máquinas industriales y en ciertos casos se ha llegado a comprobar que consumen alrededor del 10% del suministro total de energía de la industria.

Debido a las condiciones de funcionamiento de los componentes de los compresores, pueden ocurrir fallos que conlleven a fallas con eventos catastróficos, a más de la inminente detención del proceso de producción (Tran, AlThobiani, & Ball,

2014). De acuerdo a encuestas realizadas a fabricantes se pudo conocer que el 76.5% de todas las fallas en compresores se originan por fallos en sus sistemas mecánicos (Saidur, Rahim, & Hasanuzzaman, 2010). Uno de los componentes que están sometidos a un desgaste frecuente son las válvulas, pues cuando estas fallan tienen alta probabilidad de desencadenar consecuencias peligrosas como ruido excesivo, escape del aire comprimido, consumo adicional de energía y causar daños graves a la máquina e incluso su detención (Guerra, C., 2013). Al respecto, surge la necesidad detectar fallos a edad temprana o cuando sean incipientes, a fin de mejorar la eficiencia de los procesos productivos, reduciendo costos y operando en condiciones de seguridad (Verma et al., 2016).

Para abordar este problema, se han desarrollado varios métodos para detectar el estado de los compresores, uno de ellos se desarrolla utilizando la metodología tradicional de aprendizaje automático que consiste en extraer las características, seleccionirlas y clasificar patrones. Para ello es importante considerar que las válvulas son elementos clave de los compresores y la detección de fallos resulta en muchos casos difícil realizar por la falta de estacionaridad y no linealidad de la señal de vibración. Al respecto Chen & Lian, (2010a) utilizan el enfoque de máquinas de vectores de soporte para efectuar un modelo de clasificación de fallos. Para el efecto recurren a la entropía de wavelet packet como variable de entrada, mostrando en sus resultados efectividad del modelo por su capacidad de generalización con respecto a los métodos tradicionales bajo condiciones de no estacionaridad y no linealidad.

Por otro lado Mahmud Ahmed, Abdusslam, Baqqar, Gu, & Ball, (2011) aplican redes neuronales y máquinas de soporte vectorial (MSV) para la detección de fallos. Estos métodos utilizan para la extracción de características y clasificación. Los resultados de su estudio muestran que las redes neuronales (específicamente las redes neuronales probabilísticas PNN) tienen mejor rendimiento que las MVS usando características extraídas en el dominio frecuencia. Por su lado las MSV mostraron mejores resultados con características en el dominio tiempo, pero no superaron los resultados de las PNN.

Otro enfoque que se ha utilizado para la detección de fallos en compresores es el aprendizaje automático utilizando árboles de decisión. Por ejemplo, en el estudio

realizado por Aravinth, Kanna, & Sugumaran, (2016) se realiza la predicción de fallos a partir de señales de vibración de un compresor alternativo, que sirvió para extraer y seleccionar características estadísticas con el fin de detectar fallos a etapas tempranas. Los resultados mostraron una precisión del 98.33% en la detección.

Otra alternativa para el monitoreo de la condición de los compresores es aplicación de modelos matemáticos considerados totalmente tradicionales. Belman-Flores, Ledesma, Barroso-Maldonado, & Navarro-Esbrí, (2015) aplican este enfoque y adicionalmente utilizan un modelo computacional para su comparación. Para el primero utilizan ocho subprocesos internos que involucran desplazamientos infinitesimales de acuerdo al movimiento del pistón; en tanto que en las redes neuronales utilizan como variables de entrada la presión de succión, temperatura de succión, presión de descarga y velocidad de rotación del compresor. Los resultados muestran valores del error medio debajo del más menos 10% para los modelos matemáticos y para el modelado con redes neuronales por debajo del más-menos 1%.

A su vez Farzaneh-Gord & Khoshnazar, (2016) suman esfuerzos en el desarrollo de la investigación de la detección de fallos en compresores. Su investigación se orienta a la detección de fallos en las válvulas de un compresor alternativo por medio de un análisis numérico del movimiento del pistón, de la dinámica de la válvula y del caudal másico. Para el análisis se utilizaron tres volúmenes de control: (i) flujo del cilindro del compresor, (ii) flujo de succión y (iii) flujo de descarga. Para el análisis de fallas en válvulas se simulaban las siguientes fallas: fallas de la placa de la válvula, desgaste de la placa y asiento y deterioro de los resortes. La simulación se validó con los resultados experimentales que determinó que los fallos en las válvulas de succión y de descarga disminuyen el caudal másico del compresor y aumentan la temperatura del gas de descarga. También se demostró que la falla en la válvula de succión es más grave que la falla en la válvula de descarga. El aporte de estos autores es muy significativo, pues orienta a determinar qué fallos son los más comunes y cómo se pueden implementar para la simulación.

Otro aporte significativo es el realizado por Guerra, (2013) que realiza el estudio de monitoreo de la condición de compresores utilizando datos de presión, temperatura y vibración, con tres modos de fallo en las válvulas; fuga de líquido, fatiga en el resorte

de la válvula y desgaste en el asiento de la válvula. Para la detección de fallos utiliza el método Bayesiano junto con los datos de entrenamiento para crear un clasificador que determine el estado de la máquina y compara con métodos analíticos. Este autor señala que el mayor porcentaje de las fallas en compresores recíprocos se originan en las válvulas y que inciden en el 50% del costo total de reparación. Como se puede evidenciar, son varios los aportes que se han dado en los últimos años, sin embargo es necesario proponer metodologías alternativas que mejoren la detección de fallos en compresores recíprocos.

2.3 Bases teóricas

2.3.1 Compresores

Salvador, (1988) define a los compresores como “máquinas que están construidas para aumentar la presión y desplazar fluidos compresibles, pasando de presión baja a otra más alta”. Estas máquinas disminuyen el volumen de una cantidad específica de aire e incrementan su presión mediante dispositivos mecánicos como se muestra en la Figura 3.

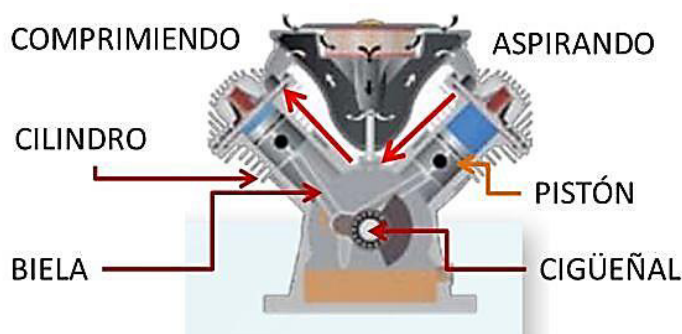


Figura 3. Estructura de un sistema de compresión. Tomado de («Diferentes tipos de compresores», 2018)

2.3.2 Tipos de compresores

Aunque todos los compresores tienen el mismo propósito, cada tipo tiene su aplicación en diferentes industrias y procesos. La principal clasificación se realiza por su principio de funcionamiento básico. De esta forma quedan divididos en dos grandes grupos, como se muestra en la Figura 4 (Instituto de Mecánica de los Fluidos e Ingeniería Ambiental, 2010). Es importante señalar que el compresor de interés para la presente investigación es de desplazamiento positivo recíprocante de simple efecto-doble etapa.

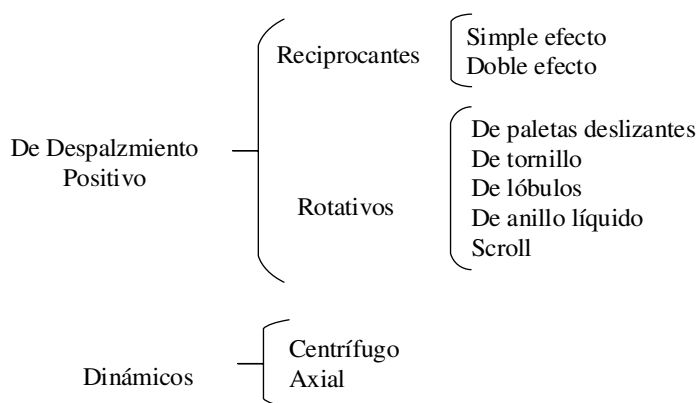


Figura 4. **Tipos de compresores.** Tomado de («Diferentes tipos de compresores», 2018).

- ***Compresor recíprocante de simple efecto doble etapa.***

En este tipo de máquinas, la compresión se obtiene por el desplazamiento del pistón que se realiza de forma lineal a través de un cilindro, como resultado se logra reducir el volumen del gas y aumentar su presión (Posada C, 2017), obteniendo potencia tres veces más que otros tipos (Ramesh, 2007). En la siguiente Figura 5 se ilustra lo descrito.

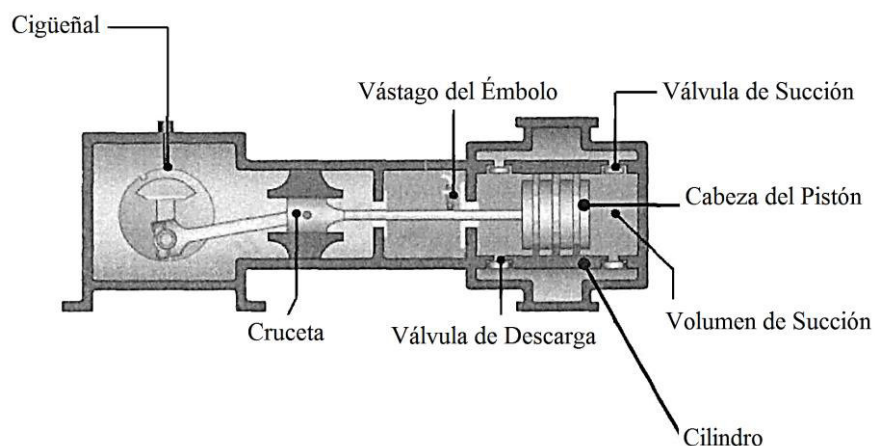


Figura 5. Configuración de un sistema de compresión recíproco. Tomado de (Ramesh, 2007).

Según Fernandez (2000), un compresor recíprocante es de simple efecto cuando trabaja sólo por un extremo del pistón (Figura 6) y es de doble etapa cuando el fluido se comprime en dos fases, la primera realiza una compresión hasta una presión intermedia de 2 a 3 bares y en la segunda alcanza sobre los 8 bares en promedio. En la Figura 7 se puede apreciar el diagrama P-V que ilustra el proceso descrito.

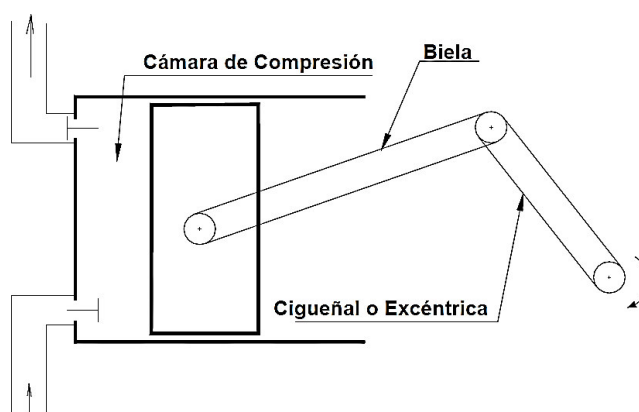


Figura 6. Esquema de un funcionamiento de compresión de simple efecto. Tomado de (Fernandez P, 2000).

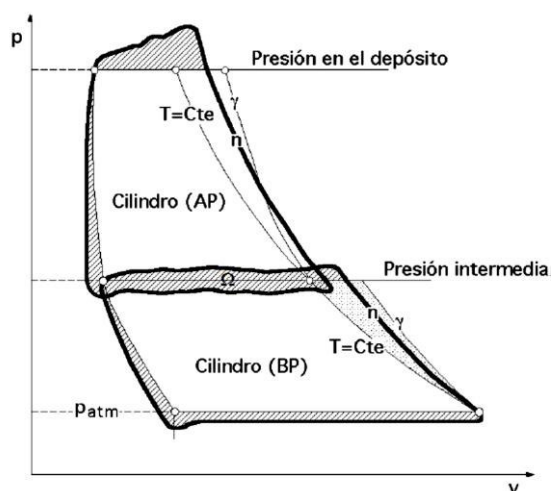


Figura 7. **Diagrama P-V de un compresor recíprocante de dos etapas.**

Tomado de (Fernandez P, 2000).

Ciclo de compresión ideal. En un compresor recíprocante la compresión ideal está determinada por cuatro etapas. En la posición 4-1 con la válvula de admisión abierta, el fluido es succionado a una presión p_1 de entrada. En la posición 1-2 se realiza la compresión del gas, disminuyendo el volumen y aumentando la presión desde p_1 hasta p_2 . Con la válvula de descarga abierta, en la posición 2-3 se expulsa el aire comprimido a la presión de descarga p_2 hasta que en la posición 3-4 existe caída brusca de presión hasta iniciar nuevamente el ciclo (Landa J, 2014). En la Figura 8 se ilustra este proceso.

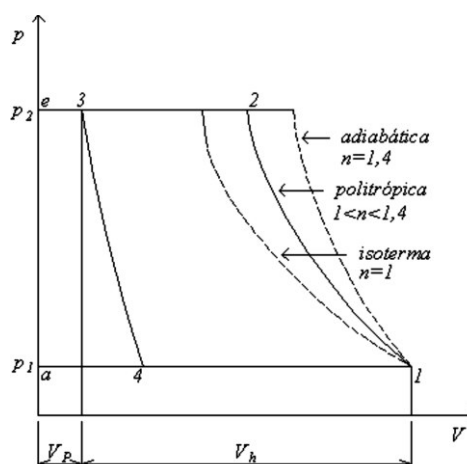


Figura 8. **Ciclo ideal de compresión.** Tomado de (Landa J, 2014).

Ciclo de compresión real. El ciclo real sufre una desviación del ciclo ideal considerando que pueden ocurrir ciertas imprecisiones en la fabricación o por desgaste en uno de los componentes del compresor, o en su defecto por el espacio que es necesario para el movimiento de apertura y cierre de las válvulas. Lo mencionado se puede apreciar en la Figura 9, donde se evidencia que el resultado de estos fenómenos da lugar a un volumen de fluido no expulsado por el pistón en la etapa 2-3 y que a su vez es expandido en la etapa 3-4 (Fernandez P, 2000).

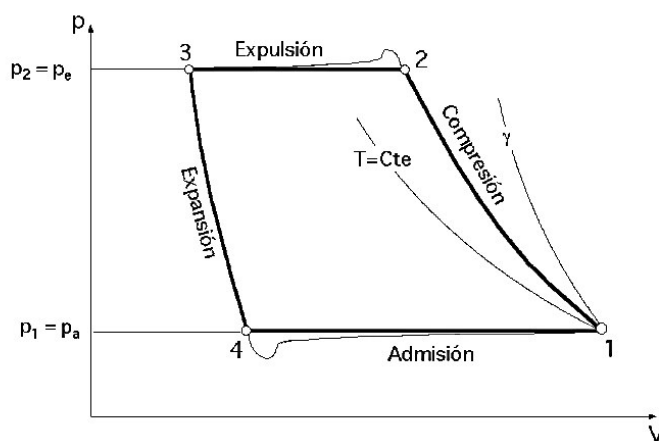


Figura 9. **Ciclo real de compresión.** Tomado de (Fernandez P, 2000)

Procesos de compresión y expansión. Los procesos 1-2 y 3-4 no son sencillos de conocer. Se suele aceptar que son politrópicos, o sea, que se cumple:

$$v^n = cte \quad [1]$$

Para algún valor de n .

Como la compresión (y también la expansión) se realiza con una masa fija de gas encerrado, durante dichos procesos se puede sustituir el volumen específico por el volumen ocupado V :

$$p \cdot V^n = cte \quad [2]$$

Si el proceso fuera adiabático, sería

$$n = k = c_p/c_v \quad [3]$$

Pero siempre hay transferencia de calor, natural o forzada. Por lo tanto, el exponente politrópico n suele tomar valores menores que k sin llegar nunca a ser $n=1$ (proceso isoterma).

2.3.3 Fallos comunes en compresores reciprocantes

Los compresores reciprocantes desempeñan un papel importante en muchos sistemas industriales y los fallos que ocurren en ellos pueden degradar el rendimiento, consumir energía adicional, causar daños graves a la máquina y posiblemente provocar el cierre del sistema (Ahmad & Kamaruddin, 2012). Una encuesta realizada a consumidores y fabricantes de compresores reciprocantes de diez países reveló que los sistemas mecánicos de los compresores causan alrededor del 76.5% de todas las fallas no planificadas (Kostyukov & Naumenko, 2016a).

Los fallos comunes del compresor recíprocante se pueden dividir principalmente en dos tipos: fallos del sistema mecánico y fallos de sistemas auxiliares (Jiang, Zhang, Jin, & Ma, 2013). En el **¡Error! No se encuentra el origen de la referencia.** se clasifican los fallos más comunes que se presentan en los compresores reciprocantes.

● Fallos en rodamientos

Los rodamientos son los componentes más importantes de las máquinas rotativas. Una de las principales causas del tiempo de inactividad de la máquina es debido a los fallos en estos elementos. El tiempo de inactividad de las máquinas

rotativas puede reducirse monitoreando la vibración y el comportamiento acústico de los elementos de la máquina (El-Thalji & Jantunen, 2015). La clasificación de fallos en rodamientos está dada por la norma ISO 15243, que divide a los fallos en seis categorías y 15 tipos de fallos. En el Cuadro 2 se presenta las categorías y los tipos de fallos (ISO 15243:2004, s. f.).

Cuadro 1. Fallos comunes en compresores reciprocantes.

Sistema	Componente	Fallos comunes
Fallos del sistema mecánico	Válvula	Fuga, atasco, fractura, asfixia
	Pistón	Abrasión del anillo de soporte, abrasión del anillo del pistón, fractura del anillo de soporte, rotura del anillo del pistón, aflojamiento de la tuerca del pistón, fuga del anillo del pistón.
	Vástago	Aflojamiento de la tuerca de seguridad de la varilla del pistón, abrasión de la varilla del pistón, fractura de la varilla del pistón.
	Cruceta	Abrasión de los zapatos de la cruceta, abrasión del pasador de la cruceta, abrasión del buje del pasador de la cruceta.
	Biela	Fractura de la biela, abrasión del buje del perno de la biela, abrasión del cojinete de la biela.
	Cigüeñal	Fractura del cigüeñal, abrasión del cojinete principal, desalineación del acoplamiento, desequilibrio.
	Tubería	Resonancia mecánica, resonancia de la columna de gas, resonancia de la columna de gas y mecanismo.
	Cilindro	Desalineación del cilindro con zapatas de cruceta, golpe de líquido, colisión en el cilindro, abrasión del cilindro.
	Partes del cuerpo	Aflojamiento de pernos de anclaje.
	Embalaje	Fugas, abrasión
Fallos de sistemas auxiliares	Fallo del sistema de enfriamiento, fallo del sistema de lubricación, fallo del sistema de filtración, aflojamiento del transductor o del circuito, posición de instalación no razonable del transductor, interferencia electromagnética.	

Fuente. Tomado de (Jiang et al., 2013).

● Fallos en válvulas

Chen y Lian, (2010) señalan que las válvulas en buen estado son vitales para que los compresores operen de manera eficiente, de la misma forma en que su estado de funcionamiento afecta directamente el desplazamiento de aire y el consumo de potencia. La dificultad de accesibilidad para la inspección de su condición ha originado el interés de investigadores sobre métodos indirectos de detección de la condición de los compresores y sus válvulas (Schantz, 2011). En los compresores reciprocantes, las válvulas de succión y descarga son los componentes en los que se producen fallos con frecuencia (Farzaneh-Gord & Khoshnazar, 2016b). En el 36% de los casos es necesario realizar una parada no programada, representando así el 50% del costo total de reparación. Esto lleva a considerar las metodologías de diagnóstico de fallos precisas, confiables y adecuadas para las válvulas, cuya finalidad es llevar a efecto un proceso productivo enmarcado en la seguridad tanto para el operario como del proceso, minimizando de esta manera los costos de mantenimiento (Tran et al., 2014), (Pichler et al., 2016).

Cuadro 2. Clasificación de fallos en rodamientos según ISO 15243:2004.

Norma	Categorías de fallos	Tipo de fallo
ISO 15243:2004	Fatiga por contacto	Inicio de fatiga en superficie - Inicio de fatiga en subsuperficie
	Desgaste	Desgaste abrasivo - Desgaste adhesivo
	Corrosión	Corrosión por humedad - Corrosión por fricción
	Erosión eléctrica	Voltaje excesivo - Fuga de corriente
	Deformación plástica	Sobrecarga - Indentación de escombros - Indentación por manejo
	Grietas y fracturas	Fractura forzada - Fractura por fatiga - Agrietamiento térmico

Fuente. Tomado de (Ortiz & Pérez, 2016).

2.3.4 *Mantenimiento de los compresores*

Bloch & Hoefner, (1996) plantean los tres niveles de mantenimiento que se pueden ejecutar en los compresores: (i) mantenimiento correctivo, (ii) mantenimiento preventivo y (iii) mantenimiento predictivo. No obstante, considerando que los compresores son de gran importancia en la industria es deseable que se lleve a efecto el mantenimiento predictivo. El mantenimiento basado en la condición (MBC) es un enfoque del mantenimiento predictivo que usa la información que se obtiene a través del monitoreo de la condición de la máquina. Si el programa se lleva correctamente, se puede reducir el costo, tareas innecesarias y riesgos asociados al mantenimiento preventivo, contribuyendo así a mejorar su productividad (S. Alaswad & Xiang, 2017), (D. Galar, Thaduri, Catelani, & Ciani, 2015), (A. K. S. Jardine, Lin, & Banjevic, 2006).

Mediante un monitoreo de la condición que tiene como parámetros las señales de vibración, se puede lograr un diagnóstico temprano para disminuir paradas de emergencia (Guiracocha, 2015). Elhaj et al., (2008) ha llevado a efecto un estudio basado en la simulación que involucra el desarrollo de un modelo matemático de cinco procesos físicos diferentes: características de velocidad-torque de un motor de inducción, variación de presión del cilindro, movimiento rotacional del cigüeñal, características de flujo a través de válvulas y vibración de las placas de válvula. Al final de su investigación concluye que las medidas de presión producen una característica de detección clara, sin embargo resalta que es una medida intrusiva y difícil de implementar. Esto orienta hacia la evaluación de parámetros que no interrumpan el normal funcionamiento de las máquinas para monitorear su condición, uno de ellos es la vibración (Potočnik & Govekar, 2016).

2.3.5 *Mantenimiento basado en la condición*

El mantenimiento basado en la condición (MBC) es una estrategia que se fundamenta en el monitoreo de la condición (MC) de los equipos y sugiere la toma de decisiones en función de información que proviene del monitoreo de parámetros como

la vibración, niveles de ruido, temperatura, entre otros. Es la estrategia con mayor atención en el ámbito científico e industrial, motivado por la capacidad de los equipos para presentar signos o señales de que va a ocurrir una falla (Ahmad & Kamaruddin, 2012). Su aplicación parte del diagnóstico a través del monitoreo de las señales, utilizando sensores u otros indicadores que revelan en tiempo real la condición del equipo (Tian, Wu, & Chen, 2014). De esta manera, el diagnóstico de fallos abarca tres aspectos: la detección, el aislamiento y la identificación. En la detección se revela el comportamiento anómalo y evidencia el momento en que se produce, el aislamiento determina dónde y el tipo de fallo que se ha producido, y la identificación establece la magnitud del fallo (Bayar, Darmoul, Hajri-Gabouj, & Pierreval, 2015). El MBC utiliza dos enfoques para el diagnóstico: (i) modelos matemáticos y (ii) modelos basados en datos (data driven models).

El principio de los modelos matemáticos se fundamenta en el conocimiento básico del proceso que es vital para poder utilizar relaciones matemáticas entre algunas variables de entrada y salida, con la finalidad de extraer información sobre cambios que puedan originar fallos (Bayar et al., 2015). Como se puede evidenciar en la Figura 10, en este enfoque, se extraen características especiales como parámetros, variables de estado o residuales que provienen de los actuadores del proceso de funcionamiento de la máquina y son adquiridos por sensores estratégicamente colocados, obteniendo de esta forma una señal para el análisis. Estas características son comparadas entre el valor nominal y el valor observado con la finalidad de determinar los posibles cambios. Para el efecto, se plantea un modelo matemático del proceso, se generan características de análisis con la aplicación de métodos de detección de cambios y se emite el diagnóstico (Isermann, 2006).

En su mayoría los sistemas se modelan bajo principios físicos utilizando relaciones matemáticas que caracterizan su funcionamiento, sin embargo el diagnóstico basado en datos proporciona mayor precisión. Los modelos matemáticos ignoran ciertos fenómenos físicos para dar mayor simplicidad al análisis, lo que le hace menos exacto al momento de realizar un diagnóstico (Belman-Flores et al., 2015). El diagnóstico basado en datos surge como una alternativa altamente efectiva, al utilizar información procedente de grandes cantidades de datos obtenidos del proceso (Kumar, Shankar, & Thakur, 2017).

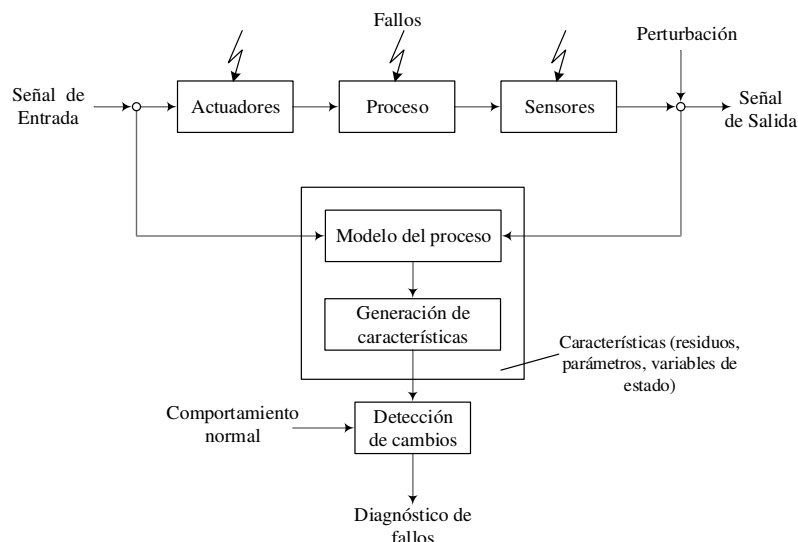


Figura 10. **Esquema general de detección de fallos basado en modelo matemático.** Adaptado de (Isermann, 2006).

Los modelos basados en datos utilizan un mapeo o reconocimiento de patrones en un espacio de medición característico (Deepam Goyal & Pabla, 2015). Toma como referencia el mapeo en estado normal y con fallo, a través del cual el sistema de diagnóstico puede reconocer si existe una desviación considerable entre los dos patrones para concluir que existe un fallo. Para compresores el modelo entrena patrones en condiciones normales y con fallo en los elementos con mayor desgaste, que en este caso son las válvulas (Aravinth et al., 2016). A continuación en la Figura 11 se muestra en esquema el proceso de modelamiento para la detección de fallos con el enfoque que se ha descrito.

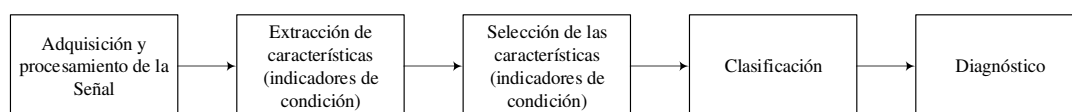


Figura 11. **Esquema de detección de fallos basado en datos.** Tomado de (Verma, Gupta, Sevakula, & Salour, 2014)

Para los compresores reciprocantes el proceso de adquisición y procesamiento de la señal comienza con la recopilación y almacenamiento de datos de parámetros como: vibración, ruido, emisión acústica entre otros (Andrew K. S. Jardine et al., 2006). Los datos adquiridos pueden contener errores que tienen que ser ajustados. El siguiente paso es el procesamiento de la señal que sirve para representar los datos en una forma de señal adecuada para su análisis e interpretación que puede ser en el dominio tiempo, frecuencia y tiempo-frecuencia (Bayar et al., 2015). El procesamiento de la señal sirve a su vez para extraer información útil que se conoce como extracción de características o indicadores de condición. El siguiente paso es la selección de las características, se trata de excluir aquellas que se consideran innecesarias o insignificantes para inferir la condición normal o de fallo de una máquina. Previo a terminar el proceso de diagnóstico, es necesario realizar la clasificación, se trata de analizar las características mediante un mapeo de la información en condiciones normales y con fallo. Originalmente este procedimiento se realiza de forma manual, sin embargo requiere un conocimiento experto y altamente calificado, volviéndose poco práctico y costoso. Una alternativa es el reconocimiento automático de patrones que se apoyan en enfoques estadísticos y de inteligencia artificial (Mahmud Ahmed, Abdusslam, Baqqar, Gu, & Ball, 2011), (Ellman & Piché, 1999), (B.-S. Yang, Hwang, Kim, & Chit Tan, 2005).

2.3.6 Mantenimiento basado en la condición frente al mantenimiento centrado en la confiabilidad.

El mantenimiento predictivo surge como alternativa para disminuir costos y asegurar la funcionalidad de los equipos, en este sentido se tienen dos estrategias dentro del mantenimiento predictivo: (i) el mantenimiento basado en la condición (MBC) y (ii) el mantenimiento centrado en la confiabilidad (MCC). Por un lado el MBC se enfoca en el monitoreo de la condición de parámetros como la vibración emisión acústica, sonido entre otras (Deepam Goyal & Pabla, 2015). Por su parte el MCC se define como el proceso que se aplica para determinar qué debe hacerse para asegurar el funcionamiento de los equipos y maquinaria. A continuación se detalla las particularidades de cada estrategia (Montilla M., Arroyave, & Silva M., 2007).

***Mantenimiento basado en la
condición***

Monitoreo de parámetros como la vibración, emisión acústica, sonido, entre otros para determinar la condición de la máquina.

Los pasos son: adquisición de datos, procesamiento de datos y toma de decisiones de mantenimiento como diagnóstico y pronóstico. Para su aplicación se utilizan dos enfoques: modelos matemáticos y modelos basados en datos.

Beneficios: Detección y diagnóstico temprano de la existencia de fallos, cuando sean aún incipientes. Intervención en los equipos sin detener su funcionamiento.

***Mantenimiento centrado en la
confiabilidad***

Identificación de las circunstancias causantes de una falla e investigación de las situaciones que lo originan.

Parte de siete preguntas para saber si existe o no fallas: ¿cuáles son las funciones?, ¿En qué forma no se pueden cumplir las funciones?, ¿En qué condiciones el equipo falla?, ¿Qué sucede cuando falla?, ¿Cuál es la probabilidad de ocurrencia de la falla? y ¿Cuál es la severidad de la falla?

Beneficios: Mayor seguridad y protección del entorno, mejores rendimientos operativos, mayor control de costos del mantenimiento, vida útil de los equipos.

2.3.7 Monitoreo de la condición

El monitoreo de la condición (MC) permite conocer el estado de la máquina a partir de signos o síntomas que provienen de un control constante de parámetros físicos de interés como velocidad angular, presión, señal de corriente del motor, análisis de vibración y emisión acústica; evitando así escenarios de operación no aceptables o paradas no programadas (Bayar et al., 2015), (Farzaneh-Gord & Khoshnazar, 2016a). En compresores las señales de mayor interés y aplicación son las señales de vibración y emisión acústica (Žilka, 2014).

● **Monitoreo de la vibración.** Para Liang et al., (2015), M. Ahmed, Baqqar, Gu, & Ball, (2012) y (Kostyukov & Naumenko, 2016), el monitoreo de la vibración es el método más aplicado. Se puede inferir que es considerado un recurso muy valioso para la detección y diagnóstico de fallos. La denominada “firma de vibración” orienta en el reconocimiento de la condición de la máquina, es así que su amplitud indica la gravedad del fallo mientras que la frecuencia podría revelar el origen (Deepam Goyal & Pabla, 2015). Las razones que dan ventaja al monitoreo de la vibración se concentran en las siguientes:

- Permite un monitoreo sostenible sin ser destructivo. En efecto, no requiere interferir en la operación de la máquina (D. Goyal & Pabla, 2016).
- El análisis de vibración puede identificar con precisión el 90% de todas las fallas de la máquina (Devendiran & Manivannan, 2016).
- Ha sido demostrado ser el método más efectivo para el monitoreo de condición por su capacidad para proporcionar un tiempo razonable antes de la falla permitiendo tomar acciones a tiempo y bajo una planificación (Hui, Hee, Leong, & Abdelrhman, 2015).

A pesar de las múltiples ventajas, la extracción de las características de la señal en ocasiones puede resultar complicado por la interferencia ocasionada por ruido proveniente de otras fuentes. Ante esto se hace necesario el desarrollo de un sistema de procesamiento y análisis de datos de alto nivel para obtener mayor certeza y objetividad en el proceso de diagnóstico (D. Goyal & Pabla, 2016). A continuación se desarrollará cada paso del proceso de detección de fallos basado en datos con orientación a compresores reciprocantes.

a) Adquisición de datos y pre-procesamiento de la señal.

El proceso de adquisición de datos se refiere a la recopilación y almacenamiento de información proveniente de elementos específicos que son de interés del investigador, implica la elección del tipo de sensor, su ubicación, cantidad, el hardware de adquisición y almacenamiento de datos, así como el procedimiento para su adquisición. (Fugate, Sohn, & Farrar, 2001). Según Jardine et al., (2006) y Bayar et al., (2015), los datos para el monitoreo se pueden clasificar en tres categorías:

1. Tipo valor.- La información adquirida se refiere a valores de parámetros como temperatura, presión, humedad.
2. Tipo onda: La representación de datos se da en series de tiempo, en esta categoría están los datos de vibración y emisión acústica.
3. Tipo multidimensional.- Los datos están contenidos en más de una dimensión, es común en imágenes.

La adquisición de datos de vibración se realiza a través de sensores que son los encargados de monitorear la vibración de un cuerpo a través de su estructura metálica, convirtiendo ésta en una señal eléctrica equivalente (Figura 12). Es de vital importancia seleccionar los sensores apropiados para que la adquisición de datos sea de confianza, puesto que la selección no sólo considera la adquisición de una señal precisa sino también su interpretación. Entre los dispositivos para la adquisición de datos de vibración están los transductores de desplazamiento, transductores de velocidad, acelerómetros y vibrómetros láser Doppler. De estos, el más utilizado es el acelerómetro por su peso ligero, frecuencia superior, rango dinámico, capacidad, robustez y alta sensibilidad (D. Goyal & Pabla, 2016).

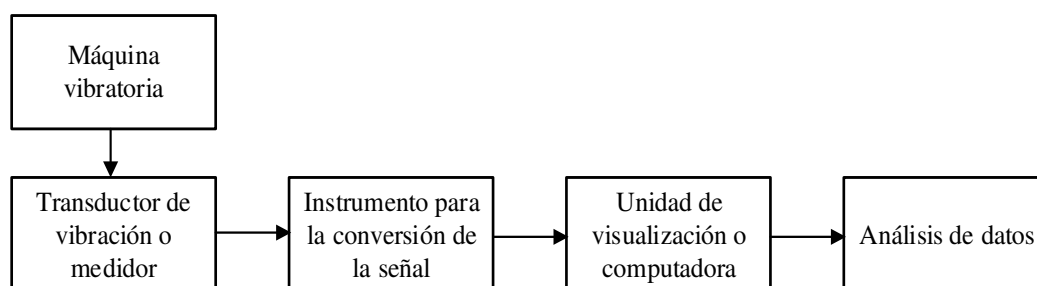


Figura 12. **Principio básico de medición de la vibración.** Tomado de (D. Goyal & Pabla, 2016).

Luego de la adquisición de datos es necesario realizar el pre-procesamiento de la señal para reducir valores atípicos y eliminar algún efecto de ruido proveniente de otros elementos. Para tal efecto se comienza con un filtrado para eliminar frecuencias

no deseadas, siendo el dominio tiempo la primera plataforma para la extracción de características útiles de los datos brutos (Verma et al., 2014). Al finalizar el filtrado, los datos se separan en dos clases: datos de condiciones normales y datos de condiciones de fallo. Para el diagnóstico sirven únicamente los primeros, así una cantidad de estos se utiliza para entrenar el modelo, mientras que el restante para su validación. Previo al diagnóstico, el modelo debe ser capaz de identificar la condición normal y de fallo de la máquina, para tal fin se utilizan diferentes técnicas como la inteligencia artificial y el aprendizaje automático (Keerqinhu et al., 2016).

b) Extracción de características

La señal adquirida por los sensores es procesada y con la aplicación de diversas técnicas se extraen características como la amplitud, frecuencia, desplazamiento, velocidad, aceleración, fase y periodo (Deepam Goyal & Pabla, 2015). También se suele utilizar características estadísticas como media, varianza, desviación estándar, curtosis, entre otras (Verma et al., 2014).

La señal de vibración que se obtiene de un compresor reciprocante contiene características no lineales, no estacionarias y acoplamiento multicomponente por factores como la holgura, rigidez, no linealidad de los cojinetes y por el movimiento de impacto en la succión y descarga de las válvulas (Haiyang, Jindong, Hui, & Yiqi, 2016). Las características extraídas o también conocidas como indicadores o parámetros de condición en los dominios tiempo, frecuencia y tiempo-frecuencia han mostrado ser útiles para el monitoreo de la condición (Mahmud Ahmed et al., 2011), sin embargo por la complejidad es necesario aplicar técnicas adecuadas y robustas para el procesamiento y análisis de la señal, entre los que se incluyen métodos estadísticos, wavelets, enfoques psico-acústicos, métodos basados en lógica neuronal y difusa, y muchos métodos modernos de aprendizaje automático (Potočník & Govekar, 2016). En la literatura se reportan tres categorías principales para el análisis de señales de señales de vibración (Guerra & Kolodziej, 2014):

- **Análisis del dominio tiempo:** Se basa en la forma de onda del tiempo, es decir calcula las características de las señales de forma de onda de tiempo como estadísticas descriptivas, entre las que se cuentan: media, desviación estándar,

estadísticas de orden superior, raíz cuadrada media, asimetría, curtosis, entre otras. Existen otros enfoques más avanzados que aplican modelos de series de tiempo hacia datos de forma de onda, su objetivo es la adaptación de los datos de forma de onda hacia un modelo paramétrico de series temporales y así extraer características (Andrew K. S. Jardine et al., 2006). No obstante, este análisis no es suficiente para extraer las características efectivas de la señal de vibración para un buen diagnóstico de fallos en compresores recíprocos (Haiyang et al., 2016).

- **Análisis del dominio frecuencia:** Se basa en la señal transformada en el dominio de la frecuencia. Este análisis tiene ventaja con respecto al análisis en el dominio tiempo por su capacidad para identificar y aislar de forma fácil el análisis de la frecuencia de los componentes de interés. Usualmente para análisis en este dominio se utiliza la transformada rápida de Fourier, que a través de una relación matemática convierte la señal que viene en el dominio del tiempo hacia el dominio frecuencia. También existen otras herramientas auxiliares para el análisis del espectro como los filtros de frecuencia, la presentación gráfica del espectro, análisis del envolvente y otras (Andrew K. S. Jardine et al., 2006).
- **Análisis del dominio tiempo-frecuencia:** Este análisis es bidimensional pues se lo realiza en el dominio del tiempo y frecuencia. Es más completo que los anteriores por su capacidad para analizar señales de forma de onda no estacionaria. Un análisis tradicional tiempo-frecuencia representa la energía o potencia de las señales de forma de onda con la finalidad de representar los patrones de fallo con mayor precisión. Las distribuciones de tiempo-frecuencia más conocidas son las transformadas de Fourier de corta duración, la distribución de Wigner-Ville y la transformada Wavelet (Andrew K. S. Jardine et al., 2006). Esta última es apropiada para modelar procesos dependientes del tiempo, pues proporcionan información global de la señal en el tiempo localizada en la frecuencia.

c) Selección de características

En el proceso de adquisición de datos se tiene gran cantidad de información que debe seleccionarse, en tal sentido se discriminan las características que no son del todo útiles y confiables para el diagnóstico (Kumar et al., 2017). Las técnicas que se utilizan

para la selección son: (i) Filtros, (ii) envolventes e (iii) integrados. Las técnicas tipo filtro se basan en la correlación y rendimiento de los clasificadores para asignar una puntuación a cada indicador, en esta técnica el procesamiento y aprendizaje de los datos se realizan de forma completamente independiente, por lo que no se tiene retroalimentación. Los envolventes a su vez se orientan a la evaluación de las características por su capacidad para la predicción, se asigna una puntuación a cada indicador para medir su rendimiento al momento de realizar el diagnóstico y al igual que el primer método no existe retroalimentación porque el procesamiento y selección son independientes. No así la técnica de integrados, pues en la selección de un subconjunto de indicadores se incorpora el entrenamiento para la construcción del modelo, convirtiéndose en la técnica más completa (Destrero, Mosci, Mol, Verri, & Odone, 2009).

d) Clasificación

La fase final del proceso de diagnóstico es la clasificación que se define como el procedimiento de mapeo de las características (indicadores) en condiciones normales y de fallo como un patrón de reconocimiento del estado de la máquina. Este procedimiento se lo ha realizado generalmente en forma manual, que requiere conocimiento experto y tiempo. Ante esto ha surgido el reconocimiento automático, basado en técnicas alternativas efectivas que utilizan modelos e inteligencia artificial aplicando redes neuronales, máquinas de soporte vectorial y algoritmos genéticos (Kumar et al., 2017), (Andrew K. S. Jardine et al., 2006). La presente investigación utiliza redes neuronales para esta etapa, por tanto a continuación se aborda el tema.

2.3.8 Aprendizaje automático (*Machine Learning*)

Sancho Caparrini (2018) define el aprendizaje automático (*machine learning*) como una rama de la inteligencia artificial que mediante técnicas basadas en modelos matemáticos dotan de aprendizaje a las computadoras. El objetivo es generar modelos que conduzcan al reconocimiento de patrones basándose en datos para generalizar el

comportamiento de algún fenómeno. Con el aprendizaje automático se logra que la computadora aprenda en base a la experiencia. En este caso, la experiencia está representada por un conjunto de datos históricos del fenómeno que se está modelando. Los datos se encuentran organizados en variables predictoras (entradas) y variables predecidas (salidas); por ejemplo, si se desea predecir la ocurrencia de un evento como la existencia de un fallo en una máquina (variable de salida), es necesario extraer ciertas características como temperatura, partículas de aceite, nivel de vibración (variables de entrada) para finalmente concluir sobre su estado.

También es importante conocer la terminología que se utiliza en el aprendizaje automático así, una característica o atributo es una variable de entrada, la clase es la identificación o etiqueta que se le asigna a las variables de salida (por ejemplo clase 0 = nivel 1 de severidad del fallo) y el clasificador es una función f que mapea el conjunto de características a una etiqueta (Fernandes de Mello & Antonelli Ponti, 2018).

- **Conjunto de datos**

El aprendizaje automático utiliza un conjunto de datos (*datasets*) obtenido del fenómeno a modelar. Se conforma por instancias que están identificadas por un conjunto de características (Cabrera, 2018). Para mejor comprensión y fácil uso de los datos se utiliza una forma tabular para expresarlos, de forma que se estructuran en filas y columnas, donde las filas denotan una instancia y las columnas representan las características. Cada característica equivale a una variable aleatoria dentro del análisis estadístico. El Cuadro 3 muestra un ejemplo de un conjunto de datos D que contiene n características con m instancias.

Previo al modelado es necesario preparar los datos para verificar las relaciones entre las distintas características así acondicionarlos para eliminar errores que se producen al momento de su adquisición. Adicionalmente se debe separar en conjuntos para las fases de creación y validación del modelo. Una de las técnicas utilizadas es el *holdout*, que consiste en la partición en datos de entrenamiento y datos de prueba. La bibliografía recomienda dividir en una proporción de 70% para el entrenamiento y 30% para la prueba; no obstante depende de la cantidad de datos y de los

requerimientos de cada caso (Na, 2017). Para el caso de modelos que requieren la optimización de hiperparámetros se recomienda la creación de un conjunto adicional conocido como conjunto de validación. También es importante puntualizar que la división debe ser aleatoria y estratificada para garantizar la diversidad en los subconjuntos. Esta acción se efectúa al subdividir el conjunto de datos en cuantas clases existan, para luego tomar aleatoriamente una cantidad adecuada por cada subconjunto (Figura 13).

Cuadro 3. Representación del conjunto de datos D

Instancia	C_1	C_2	...	C_j	...	C_n
D_1	V_{11}	V_{12}	...	V_{1j}	...	V_{1n}
D_2	V_{21}	V_{22}	...	V_{2j}	...	V_{2n}
...
D_i	V_{i1}	V_{i2}	...	V_{ij}	...	V_{in}
...
D_m	V_{m1}	V_{m2}	...	V_{mj}	...	V_{mn}

Fuente. Adaptado de (Cabrera, 2018).

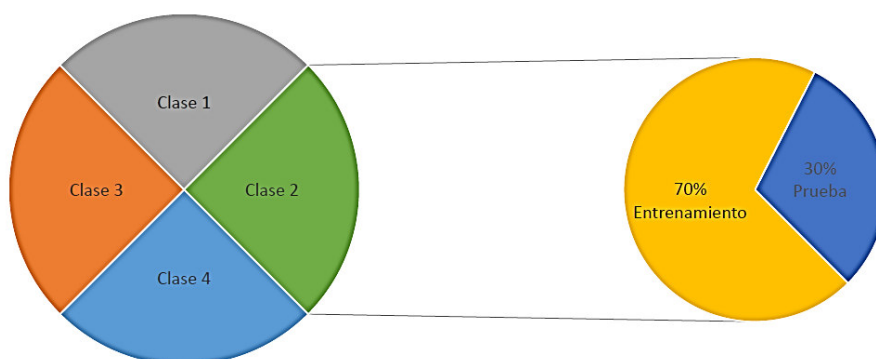


Figura 13. Aplicación de houldout para un conjunto de cuatro clases. Tomado de (Cabrera, 2018).

- **Tipos de aprendizaje**

El aprendizaje automático se divide en dos grandes grupos: (i) *no supervisado* y (ii) *supervisado*.

El *aprendizaje supervisado* se basa en una correspondencia entre las entradas y las salidas deseadas, de forma que se tiene conocimiento de las etiquetas que se le asignan al conjunto de variables de entrada que caracterizan una instancia. En este tipo de aprendizaje el objetivo es aproximar al mejor clasificador posible $f: x \rightarrow y$, siendo x la variable de entrada y y la variable de salida en un espacio de características determinado. En la Figura 14 se muestra una tarea de aprendizaje con dos clases, la línea que divide se llama frontera de decisión y se forma por datos que tienen la misma probabilidad de pertenecer a una clase u otra. También se puede apreciar que existen puntos que están en la clase equivocada, esto significa que el modelo no es óptimo y es necesario realizar un ajuste; la diferencia entre el valor obtenido y el valor deseado se llama función de pérdida, error o costo. El objetivo final es obtener un modelo con el menor error posible y esto se logra con técnicas de optimización.

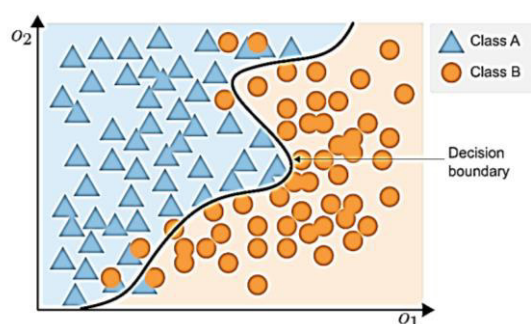


Figura 14. **Aprendizaje supervisado de dos clases.** Tomado de (Virtanen, Plumbley, & Ellis, 2017).

La estructura formal del aprendizaje supervisado se muestra a continuación (Carbonell et al., 1983), (Cabrera, 2018).

$$\hat{y} = f(x), \forall (x, y) \in D \quad [4]$$

Donde se denota la correspondencia entre la variable observada x y la variable deseada y , en un conjunto de datos D . De modo que la predicción \hat{y} producto del aprendizaje f debe ser lo más próxima al valor deseado.

Por otro lado, en el aprendizaje *no supervisado* se asignan características sin la existencia de una etiqueta que permita al algoritmo clasificar las instancias. Su objetivo es agrupar por similitud sin la posibilidad de definir cómo se comporta cada instancia. Por tanto no existe una correspondencia $x \rightarrow y$ y a diferencia del aprendizaje supervisado, se procede a un agrupamiento por clases y no por etiquetas. La Figura 15 ilustra un ejemplo de lo descrito.

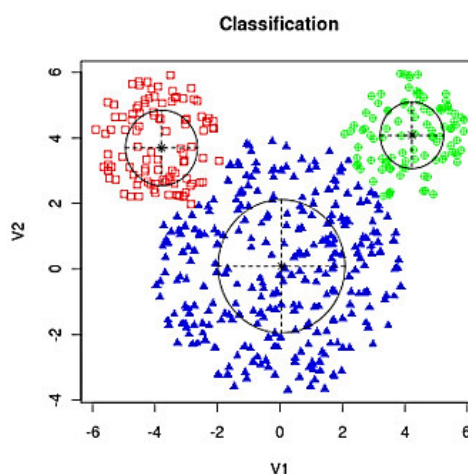


Figura 15. **Clasificación por aprendizaje no supervisado.** Tomado de (Fernando Sancho Caparrini, 2018).

- **Medición de la eficiencia del aprendizaje**

Un modelo de aprendizaje automático se mide por su capacidad para generalizar el comportamiento aprendido en la fase de entrenamiento. Una de las formas de hacerlo es la matriz de confusión que se basa en una representación visual de los

valores obtenidos por el modelo frente a los valores esperados. Para entender mejor lo expuesto se muestra en el Cuadro 4 un ejemplo de matriz de confusión para tres clases. Donde, el valor que se encuentra en la primera fila y tercera columna indica que 4 muestras fueron clasificadas como clase 3 pero pertenecen a la clase 1. También se puede observar que 275 muestras fueron clasificadas correctamente en la clase 3 (TP), 7 muestras fueron asignadas erróneamente a la clase 3 (FP) y 5 muestras que pertenecen a la clase 3 no fueron asignadas a ella (FN).

Cuadro 4. Matriz de confusión para tres clases

	Predicción clase 1	Predicción clase 2	Predicción clase 3
Valor real clase 1	256	2	4
Valor real clase 2	1	250	3
Valor real clase 3	3	2	275

Fuente. Adaptado de (Cabrera, 2018)

A partir de esta matriz es posible calcular ciertas métricas para la clasificación multiclase. A continuación se presentan algunas, teniendo presente que TP_i es un verdadero positivo para una determinada clase c_i , TN_i es un verdadero negativo, FP_i es un falso positivo, FN_i es un falso negativo, l es la totalidad de clases y M es el promedio macro (Loja & Vinicio, 2017), Cabrera, (2018).

- Verdaderos positivos (TP). Corresponde a las muestras que fueron clasificadas correctamente a una clase. Para una clase específica c el valor se calcula como se indica a continuación.

$$TP(c) = M(c, c) \quad [5]$$

- Verdaderos negativos (TN). Es el número de muestras que fueron clasificadas correctamente como no pertenecientes a una clase.

$$TN(c) = \sum_{i=1, i \neq c}^n M(c, c) \quad [6]$$

- Falsos positivos (FP). Es el número de muestras que fueron clasificadas erróneamente a una clase.

$$FP = \sum_{i=1, i \neq c}^n M(i, c) \quad [7]$$

- Falsos negativos (FN). Es el número de muestras que perteneciendo a una clase no fueron asignadas a ella.

$$FP = \sum_{i=1, i \neq c}^n M(c, i) \quad [8]$$

- Exactitud promedio: Eficacia medida de un clasificador.

$$AA = \frac{\sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + FN_i + TN_i}}{l} \quad [9]$$

- Tasa de error: Promedio del error de clasificación.

$$ER = \frac{\sum_{i=1}^l \frac{FP_i + FN_i}{TP_i + FN_i + FP_i + TN_i}}{l} \quad [10]$$

- Precisión: Promedio por clase de la coincidencia de las etiquetas de las clases de datos con las del clasificador.

$$Precisión = \frac{\sum_{i=1}^l \frac{FP_i}{TP_i + FP_i}}{l} \quad [11]$$

- Recall: Promedio por clase de la eficacia de un clasificador para etiquetas positivas de las clases.

$$Recall = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + FN_i}}{l} \quad [12]$$

- Fscore: Relación entre las etiquetas positivas de los datos y aquellas conseguidas por un clasificador.

$$Fscore = \frac{(\beta^2 + 1)Precisión_M Recall_M}{\beta^2 Precisión_M + Recall_M} \quad [13]$$

2.3.9 Redes neuronales artificiales

Las redes neuronales artificiales (RNA) son modelos computacionales corresponden basados en el principio de funcionamiento del cerebro humano. Su estructura se conforma por conexiones con pesos (parámetros) hacia procesadores (neuronas) que tienen la función de clasificar información a través de aproximadores para el reconocimiento de patrones. Esto resulta posible porque son capaces de aprender, memorizar y clasificar información a través de la experiencia desarrollada en una etapa de aprendizaje (Henao, 2000). El uso de redes neuronales artificiales para realizar el mapeo de características que definen la condición de una máquina es actualmente la técnica que mayor interés ha despertado, por ser una alternativa en la solución de casos donde una salida computacional precisa no es factible o cuando se requiere de algoritmos muy complejos. (Cruz, 2011).

Como se puede apreciar en la Figura 16, las neuronas biológicas están formadas por un cuerpo celular o soma, de donde parte una rama primaria o axón, llegando a un árbol de ramificaciones secundarias compuesto por dentritas (López & Fernández, 2008). De esta forma, se recepta la información a través de ellas y una vez que ingresa, el soma actúa como un órgano de cómputo que procesa la información emitiendo señales como estímulos nerviosos (neurotransmisores), para finalmente enviarlos por

medio del axón una salida (potenciales de acción) hacia otras neuronas. La conexión entre el axón de una neurona y las dendritas de otra recibe el nombre de sinapsis y define el tipo de relación entre ellas.

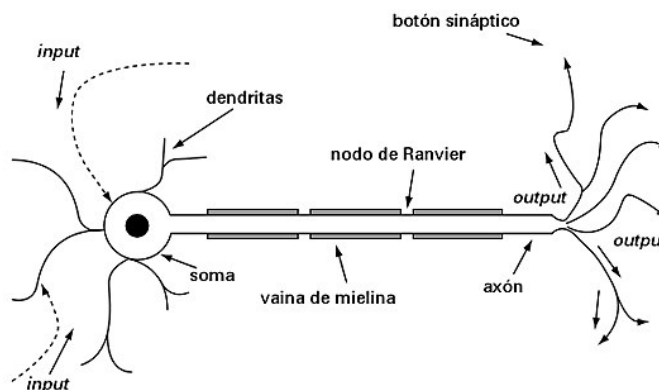


Figura 16. **Estructura de una neurona biológica.** Tomado de (Lahoz-Beltrá, 2004).

A su vez las neuronas artificiales presentan una configuración que se ilustra en la Figura 17. De forma que en una determinada neurona ingresan señales (x) provenientes de otras neuronas cada una con un peso sináptico (w), que denota la incidencia sobre cada entrada; estas se suman junto con un valor de umbral llamado bias (b) para pasar a una función de activación (f) que finalmente genera una salida (y) (Hagan et al., 2014). Partiendo de esta configuración, los modelos matemáticos que resultan para una sola entrada y múltiple entrada respectivamente son los siguientes:

$$y = f(w \cdot x + b) \quad [14]$$

$$y = \sum f(w_n \cdot x_n + b_n) \quad [15]$$

De esta relación se deduce que, al igual que en las neuronas biológicas que requieren un impulso para la activación (neurotransmisores), las neuronas artificiales

utilizan una *función de activación* que toma la información de entrada total y le combina con el estado inicial para generar un nuevo estado de activación (López & Fernández, 2008). Entre las funciones que se reporta en la bibliografía como más utilizadas (Henao, 2000) están la función lineal, función escalón, función mixta o lineal por partes, función sigmoidea, función gaussiana, función sinusoidal y función tangente hiperbólica. Su aplicación depende del tipo de problema que se pretende resolver.

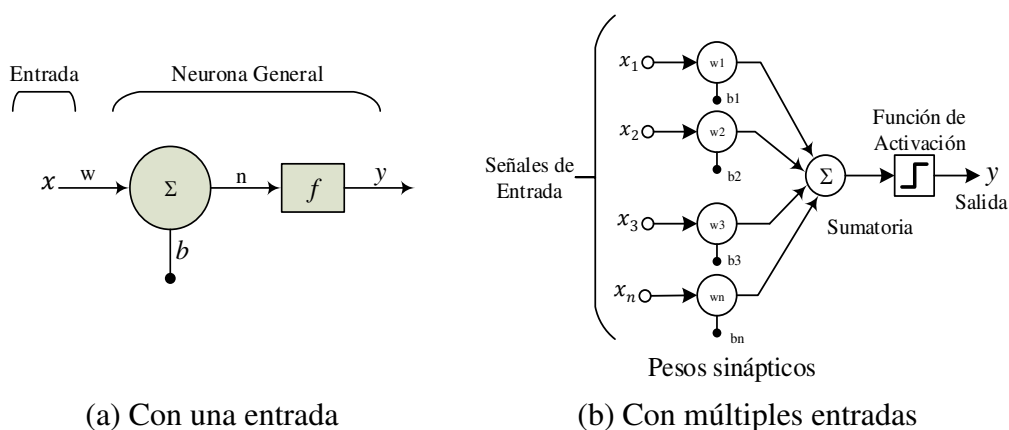


Figura 17. **Estructura de una neurona artificial.** Adaptado de (Hagan, Demuth, & Beale, 2014).

- **Arquitectura de una red neuronal artificial.**

Se conoce como arquitectura a la forma en que se conectan las neuronas. Cada neurona recibe señales de alimentación de los nodos de entrada, los procesa y emite una salida, de esta forma la red se estructura por capas. Las capas de entrada contienen la información del mundo exterior, las capas ocultas o intermedias son las que procesan y suministran una nueva entrada a las siguientes capas y por último, la capa de salida se encarga de proporcionar el resultado del aprendizaje de la red (Figura 18). Cada capa se compone de una matriz de pesos, de un vector de entrada, de la función de transferencia y del vector de salida. La conexión entre nodo y nodo se caracteriza por

la forma en que se relacionan mediante la matriz de pesos, donde las columnas representan los pesos entre las diferentes capas y las filas las instancias. Las entradas de la red corresponden a los datos a ser procesados, a su vez las salidas identifican las variables que se busca del procesamiento de la red. La cantidad de nodos en las capas ocultas depende de la complejidad de la función a ser estimada entre la capa de entrada y la capa de salida (Buscema, 2013). Por otro lado, el tipo de problema a solucionar, determina la arquitectura. Para el efecto existen dos topologías utilizadas frecuentemente: redes de propagación hacia adelante y redes recurrentes. Las primeras, reciben señales de capas anteriores y transmiten información únicamente hacia adelante (*feedforward networks*), caracterizándose además por contar con múltiples capas; en tanto que las redes recurrentes (*feedback networks*) tienen conexiones de retroalimentación que supone una evolución hacia un estado estable (López & Fernández, 2008).

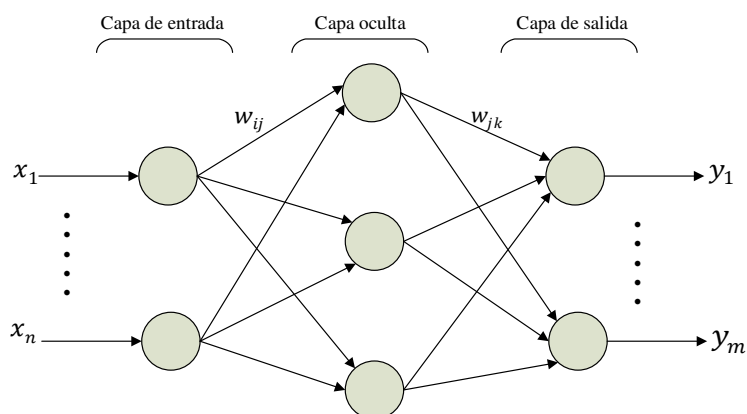


Figura 18. **Esquema de una red neuronal.** Adaptado de (Henao, 2000).

Otro aspecto importante a tratar en las redes neuronales es su aprendizaje. Según Henao, (2000) el aprendizaje es el proceso mediante el cual se realiza la configuración de la red para obtener las salidas deseadas a partir del fortalecimiento de los pesos sinápticos. Para el efecto se utilizan múltiples iteraciones de ajustes de pesos de modo que la red sea capaz de responder de forma distinta y mejorada a los estímulos del medio. Luego de cada iteración la red retorna estimaciones de la salida para cada

instancia ingresada. Este resultado se compara con las salidas deseadas y la diferencia entre ellas se denomina error, siendo el objetivo del aprendizaje minimizarlo.

2.3.10 Redes neuronales recurrentes

Las redes neuronales tradicionales (feedforward) se utilizan en muchas aplicaciones por su capacidad de aproximación universal de funciones. No obstante, cuando se requiere modelar sistemas dinámicos donde sus variables predictoras y/o variables de salida son función del tiempo no son del todo útiles por su arquitectura sin ciclos. En su lugar surgen las redes recurrentes que contienen en su estructura realimentaciones que pueden darse entre neuronas de diferentes capas, neuronas de una misma capa o en su defecto a sí misma. Precisamente ésta característica dota a la red la capacidad de almacenar información de eventos pasados para generar un nuevo estado en las siguientes capas (Bianchi, Maiorino, Kampffmeyer, Rizzi, & Jenssen, 2017) y es justamente lo que ha motivado su uso en la aproximación de sistemas dinámicos no lineales como la detección de fallos, pues los datos de vibración se reflejan en series de tiempo.

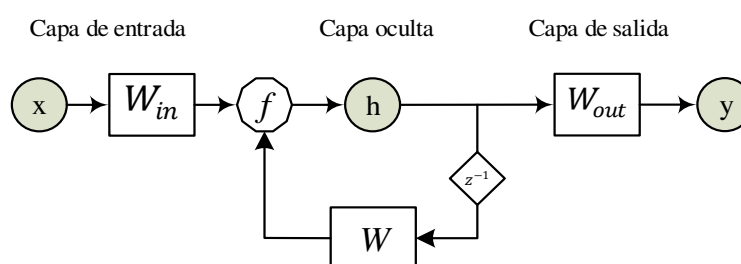


Figura 19. **Esquema de una red neuronal recurrente simple.** Tomado de (Bianchi et al., 2017).

El esquema ilustrado en la Figura 19 corresponde a la configuración de una red recurrente simple. Los círculos muestran los nodos de la capa de entrada x , la capa

oculta h o estado de la red y la capa de salida y respectivamente, W son las matrices que contienen los pesos sinápticos en cada capa, f indica la transformación no lineal realizada por las neuronas y z^{-1} es el valor de retardo. El objetivo en el proceso de aprendizaje es optimizar los pesos y de esa manera minimizar la función de pérdida producto de la diferencia entre los valores de la predicción $\hat{y}(t)$ y los valores deseados $y(t)$.

La arquitectura que se establezca determina la forma en que la información fluye a través de las capas. Existen varias formas de redes recurrentes, sin embargo las más sobresalientes son las redes Jordan y las redes Elman (Figura 20). Las primeras cuentan con una retroalimentación desde la capa de salida hacia las unidades contextuales, pudiendo también contener bucles sobre sí mismas, de forma que las unidades contextuales actúan en paralelo con la capa de entrada y la realimentación que reciben representa un nuevo estado de la red.

Por otro lado las redes Elman se proponen como una alternativa para sistemas de longitud creciente, pues las redes Jordan presentan cierta dificultad para esta tarea. Es así que la topología de las redes Elman detalla una estructura con realimentación desde las capas ocultas a la capa de contexto (Cabrera, 2018).

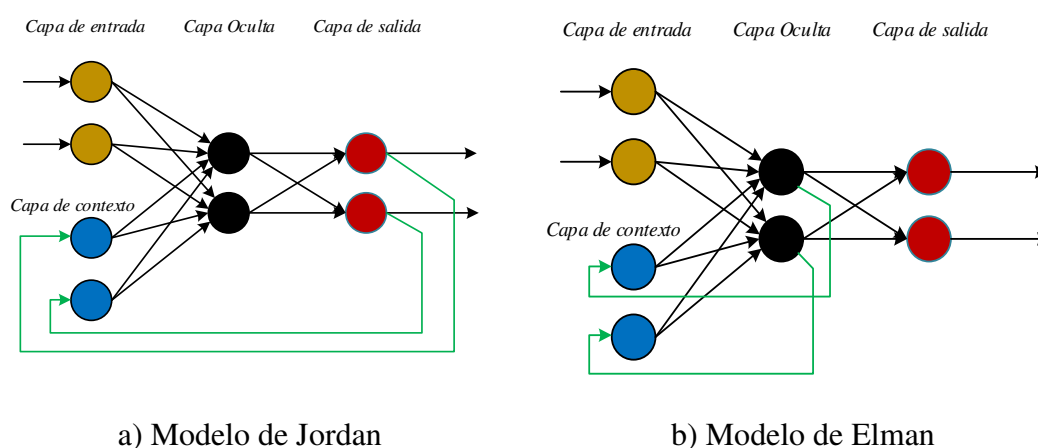


Figura 20. **Modelos de redes neuronales recurrentes.** Adaptado de (Cabrera, 2018)

La descripción formal de las redes recurrentes se efectúa en forma matricial y consta de un vector $\mathbf{x}(t)$ de entrada que contiene cada una de las variables de entrada, éstas a su vez pasan a un estado de activación representado por el vector $\mathbf{h}(t)$ en la capa oculta para luego ser procesados nuevamente y obtener salida $\hat{\mathbf{y}}(t)$. Es importante que recordar que cada capa se relaciona con los pesos sinápticos W . En el modelo de Elman la relación matemática que representa a esta red es:

$$\mathbf{h}(t) = f(W_{in}\mathbf{x}(t) + W\mathbf{h}(t - 1)) \quad [16]$$

$$\hat{\mathbf{y}}(t) = g(W_{out}\mathbf{h}(t)) \quad [17]$$

Para el modelo de Jordan se utiliza la expresión:

$$\mathbf{h}(t) = f(W_{in}\mathbf{x}(t) + W_{fb}\hat{\mathbf{y}}(t - 1)) \quad [18]$$

$$\hat{\mathbf{y}}(t) = g(W_{out}\mathbf{h}(t)) \quad [19]$$

Donde $\mathbf{x}(t) \in \mathbb{R}^{N_{in}}$, $\mathbf{h}(t) \in \mathbb{R}^{N_h}$ y $\hat{\mathbf{y}}(t) \in \{0,1\}^{N_{out}}$. N_{in} es el tamaño del vector de estímulo de entrada (capa de entrada), N_h es el tamaño del vector de estado (capa oculta) y N_{out} es el tamaño del vector de salida (capa de salida); todos estos en un instante t . También, es importante puntualizar que en ambos modelos tanto f como g son funciones no lineales. Del modelo descrito en la ecuación 16 se determina que el vector $\mathbf{h}(t)$ contiene la información del estado previo $\mathbf{h}(t - 1)$ que está ponderado por los pesos sinápticos de $W \in \mathbb{R}^{N_h \times N_h}$ y a esto se suma la información del vector de entrada $\mathbf{x}(t)$ que también está ponderado por el peso sináptico $W_{in} \in \mathbb{R}$. El vector $\hat{\mathbf{y}}(t)$ está modelando un proceso Multinoulli con variable aleatorias exclusivas de N_{out} representadas por el vector $\mathbf{y}(t)$. Como resultado las ecuaciones 17 y 19 representan el caso de regresión de Softmax con $W_{out} \in \mathbb{R}^{N_{out} \times N_h}$ parámetros. Por su lado en el modelo de Jordan presentado en la ecuación 19 se tiene alimentación de la salida previa $\hat{\mathbf{y}}(t - 1)$ ponderada por $W_{fb} \in \mathbb{R}$ al vector de estado $\mathbf{h}(t)$. Normalmente la función $f(\cdot)$ es la tangente hiperbólica descrita por $\tanh(x) = \frac{\exp x - \exp -x}{\exp x + \exp -x}$ para limitar los valores en el vector de estado.

El objetivo de entrenamiento de las redes consiste en optimizar los pesos sinápticos y los bias asociados a ellos. Para el efecto el método más común es la regla del descenso del gradiente que se basa en la minimización de la función error $E(y, \hat{y})$ que está definida por la diferencia entre los valores de predicción y los valores reales. El objetivo de este método es encontrar un mínimo lo suficientemente bueno de esta función con el uso del vector gradiente como información de actualización de pesos. Para la optimización se inicia con un vector de pesos, a partir del cual se genera otro que tiene nuevos pesos para alcanzar un valor de convergencia (Figura 21). Este proceso se vuelve repetitivo hasta que la tolerancia del error sea aceptable.

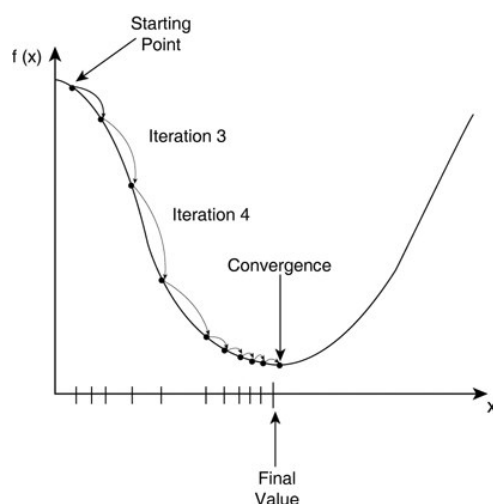


Figura 21. **Descenso del gradiente.** Tomado de (F. Sancho, accedido 2018).

En el caso de las redes neuronales recurrentes el entrenamiento se basa en el ajuste del error entre el instante de tiempo actual y el inicial. Desde un punto de vista práctico la aplicación del método de descenso del gradiente involucra pérdida de información a largo plazo (Bengio, Simard, & Frasconi, 1994) por lo tanto no resulta adecuado para sistemas con dinámica extendida. El estudio de la condición de los compresores se considera como tal, pues la vibración se representa en el dominio tiempo con relación temporal a largo plazo. De acuerdo a esto, en el presente estudio se aplicará las redes neuronales recurrentes LSTM (Long Short Term Memory) que

fueron creadas por Hochreiter y Schmidhuber en 1997 y se caracterizan por contar con celdas de memoria que son unidades formadas por nodos con un modo de conectividad específico, permitiendo a la red recordar información a corto y largo plazo en función de sus entradas. La memoria a largo plazo se produce por el valor de los pesos que se modifican lentamente en el proceso de aprendizaje dando un conocimiento global de los datos, y la memoria a corto plazo está definida por activaciones efímeras que pasan entre nodos sucesivos (Lipton, Berkowitz, & Elkan, 2015). La estructura de una red LSTM se muestra en la siguiente Figura 22.

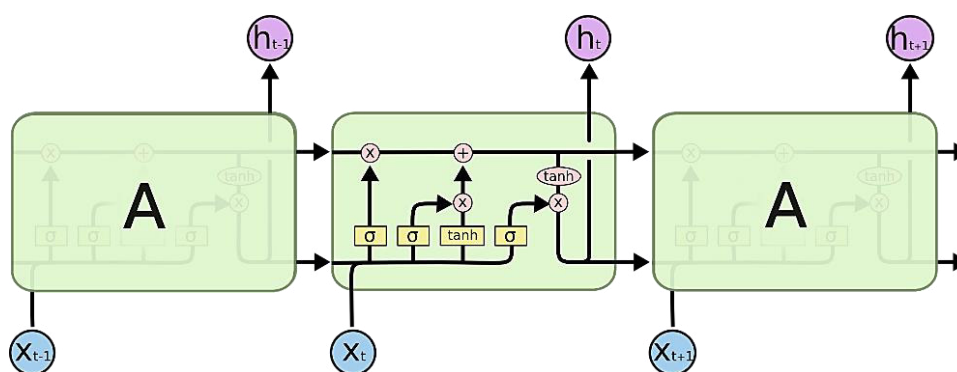


Figura 22. **Estructura de una red LSTM.** Tomado de («Understanding LSTM Networks -- colah's blog», accedido el 2018).

La peculiaridad que tienen para agregar o desechar información en cada estado, está regulado por estructuras llamadas compuertas que controlan el flujo de información a través de la red. Así, la compuerta de entrada regula la información que puede ingresar a la red, la compuerta de la capa oculta controla la información que se almacena, y la compuerta de salida define cuándo se utiliza dicha información. Esta particularidad convierte a las redes LSTM en una herramienta muy fiable para la detección de fallos, pues su rendimiento en el aprendizaje de secuencias ha sido alto (Bianchi et al., 2017). Al respecto se han aplicado en diferentes áreas como en el procesamiento del lenguaje natural, en el reconocimiento de escrituras (Hochreiter & Schmidhuber, 1997), reconocimiento de voz (Graves, Mohamed, & Hinton, 2013), reconocimiento de gestos e imágenes (Vinyals, Toshev, Bengio, & Erhan, 2017) y para

predecir series de tiempo de carga eléctrica (Mandal, Senjyu, Urasaki, & Funabashi, 2006). Como se vio anteriormente la red neuronal recurrente LSTM es un modelo computacional derivado de las redes recurrentes capaz de aprender dependencias a largo plazo. Para esto el vector de estado se calcula a través de un complejo mecanismo para olvidar y actualizar la información formalizada por las siguientes ecuaciones:

$$\tilde{\mathbf{c}}(t) = \sigma(W_{ch}\mathbf{h}(t-1) + W_{ci}\mathbf{x}(t)) \quad [20]$$

$$\mathbf{z}(t) = \sigma(W_{zh}\mathbf{h}(t-1) + W_{zi}\mathbf{x}(t)) \quad [21]$$

$$\mathbf{r}(t) = \sigma(W_{rh}\mathbf{h}(t-1) + W_{ri}\mathbf{x}(t)) \quad [22]$$

$$\mathbf{c}(t) = \mathbf{z}(t) \odot \mathbf{c}(t-1) + \mathbf{r}(t) \odot \tilde{\mathbf{c}}(t) \quad [23]$$

$$\mathbf{s}(t) = \sigma(W_{sh}\mathbf{h}(t-1) + W_{si}\mathbf{x}(t)) \quad [24]$$

$$\mathbf{h}(t) = \mathbf{s}(t) \odot \tanh(\mathbf{c}(t)) \quad [25]$$

Donde se puede identificar cuatro redes neuronales en las ecuaciones 20, 21, 22 y 24 con tuplas de entrada y pesos ocultos ($W_{ci} \in \mathbb{R}^{N_h \times N_{in}}, W_{ch} \in \mathbb{R}^{N_h \times N_h}$), ($W_{zi} \in \mathbb{R}^{N_h \times N_{in}}, W_{zh} \in \mathbb{R}^{N_h \times N_h}$), ($W_{ri} \in \mathbb{R}^{N_h \times N_{in}}, W_{rh} \in \mathbb{R}^{N_h \times N_h}$) y ($W_{si} \in \mathbb{R}^{N_h \times N_{in}}, W_{sh} \in \mathbb{R}^{N_h \times N_h}$) de cada red neuronal respectivamente. El elemento $\sigma(\cdot)$ es la función logística que está descrita por $\sigma(x) = \frac{1}{1+\exp(-x)}$. A su vez, la ecuación 23 detalla el vector de estado $\mathbf{c}(t)$ de la celda que resume la información a corto y largo plazo a través del tiempo. Por otro lado la ecuación 24 representa el vector $\mathbf{s}(t)$ que selecciona la información que considera relevante de $\mathbf{h}(t)$, el vector $\mathbf{z}(t)$ actúa en el proceso de actualización del vector de estado de la celda junto con el vector $\mathbf{r}(t)$ que reemplaza la información olvidada. El vector $\tilde{\mathbf{c}}(t)$ es la nueva información que reemplaza a la que se olvidó y \odot es el operador de producto Hadmard (representa el producto término a término de dos matrices de igual tamaño). Estos términos sirven para realizar un modelo de aprendizaje profundo, pudiendo apilar progresivamente capas ocultas (h_1, h_2, \dots, h_{nl}) que robustezcan el modelo generado.

2.3.11 Optimización Bayesiana (OB)

El método de OB surge como una alternativa para el ajuste de modelos (Shahriari, Swersky, Wang, Adams, & Freitas, 2016). Esto se logra optimizando una función objetivo $f(\mathcal{M}_\phi)$ que se representa matemáticamente como:

$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} f(\mathcal{M}_\phi) \quad [26]$$

Donde $\hat{\phi}$ es el conjunto de hiperparámetros que minimizan la función f con un modelo \mathcal{M}_ϕ entrenado con esos hiperparámetros. De esta manera para efectuar la optimización se definen dos funciones: la función objetivo $f(\mathcal{M}_\phi)$ y la función de adquisición $S(\phi, p(\text{error}|\phi))$. La primera evalúa el rendimiento del modelo parametrizado por ϕ y la segunda se encarga de explorar un espacio de búsqueda con áreas más favorables para la optimización. Es oportuno explicar que $p(\text{error}|\phi)$ es la probabilidad del error para el conjunto de hiperparámetros en el espacio de búsqueda. El proceso se efectúa de la siguiente forma:

- 1) Inicializar el modelo $p(\text{error}|\phi)$
- 2) Obtener un nuevo conjunto de hiperparámetros a través de la resolución de $\hat{\phi} = \operatorname{argmin}_\phi S(\phi, p(\text{error}|\phi))$.
- 3) Evaluar $\hat{\phi}$ utilizando $f(\mathcal{M}_{\hat{\phi}})$
- 4) Agregar el resultado a la base de conocimiento $\mathcal{D} = \mathcal{D} \cup (\hat{\phi}, f(\mathcal{M}_{\hat{\phi}}))$
- 5) Estimar un nuevo modelo $p(\text{error}, \mathcal{D})$ basado en la información de \mathcal{D} .
- 6) Repetir el procedimiento desde el paso 2 hasta un número predeterminado de iteraciones.

La función $f(\cdot)$ dependerá del modelo específico donde se aplica los hiperparámetros, y para la estimación del modelo $p(\text{error}|\phi)$ existen varias propuestas como Random Forest o Procesos Gaussianos, sin embargo para esta investigación se utiliza el método de estimadores Parzen (Bergstra, Yamins, & Cox,

2013). La estimación de $p(error|\phi)$ se efectúa a partir de un modelo generativo $p(error|\phi)$ según la regla de Bayes:

$$p(error|\phi) = \frac{p(error|\phi)p(error)}{p(\phi)} \quad [27]$$

Donde el modelo generativo $p(error|\phi)$ está compuesto por las funciones de densidad $l(\phi)$ y $g(\phi)$ obtenidas de instancias en \mathcal{D} con su evaluación $f(\cdot)$ menor que el límite del error y mayor o igual al error respectivamente.

$$p(error|\phi) = \begin{cases} l(\phi) & error < error^* \\ g(\phi) & error \geq error^* \end{cases} \quad [28]$$

Para la función $S(\cdot)$ generalmente se utiliza el mejor negativo esperado que con la ecuación 28 resulta en la siguiente combinación:

$$NEI_{error^*}(\phi) \propto \left(\gamma + \frac{g(\phi)}{l(\phi)} (1 - \gamma) \right)^{-1} \quad [29]$$

Donde γ es el valor más bajo de $f(\mathcal{M}_{\hat{\phi}})$ en la base conocimiento \mathcal{D} .

2.3.12 Marcos conceptuales o glosario

Aprendizaje automático	Rama de la inteligencia artificial que tiene como objetivo de enseñar el aprendizaje de las máquinas por medio de algoritmos (Mendieta, 2018).
Compresores:	Máquinas que tienen por finalidad aportar energía a los fluidos compresibles (gases y vapores) sobre los que operan, para hacerlos fluir aumentando al mismo tiempo su presión (Fernandez P, 2000).

Diagnóstico:	Trata de la detección, aislamiento e identificación de fallos cuando se produzcan (Valenzuela & César, 2008).
Doble etapa	Consiste en dos unidades pistón-cilindro montadas en serie en el mismo compresor las cuales comprimen el gas en dos etapas sucesivas (Posada C, 2017).
Espectro de vibración:	Es una imagen de datos obtenida en el lugar donde se ha ubicado el sensor (acelerómetro) en la máquina. Generalmente indica la frecuencia y la amplitud (Liu & Hu, 2014) .
Falla:	Se produce cuando un elemento mecánico ha dejado de funcionar Valenzuela & César, 2008).
Fallo:	Característica del elemento que no se encuentran en condiciones óptimas de funcionamiento Valenzuela & César, 2008).
Mantenimiento basado en la condición:	Es un proceso que permite conservar a la máquina en buen estado, prediciendo lo que pueda pasar eventualmente en un futuro con sus componentes mediante técnicas monitorizadas (Sánchez, 2017).
Máquinas reciprocantes o alternativas:	Son aquellas que transforman un movimiento netamente lineal en un movimiento rotacional, o viceversa, su operación está basada en el mecanismo biela manivela (Ramesh, 2007) .

Monitoreo de condición:	Evalúa el estado de la maquinaria detectando los fallos a tiempo, implementando acciones de mantenimiento necesarias (Andrew K. S. Jardine et al., 2006) .
Perturbación	Entrada desconocida e incontrolada que actúa en el sistema (Isermann, 2006).
Red Neuronal	Aproximador universal que se basa en el principio de funcionamiento de una neurona biológica (Henao, 2000).
Residuo	Indicador de fallo, basado en las desviaciones entre las mediciones y el cálculo de la ecuación del modelo (Isermann, 2006).
Señal	Perturbación utilizada para transmitir información a través de un sistema de comunicación (Carbonell et al., 1983).
Simple efecto	Cuando la etapa compresión se realiza solamente a un lado del pistón, generalmente al lado opuesto de la biela (Bloch & Hoefner, 1996).
Sistema	Conjunto de elementos que interaccionan y que forman una unidad para la comprensión de un fenómeno (Mendieta, 2018).
Sistema Dinámico	Sistema donde se produce un cambio en el tiempo (Cabrera, 2018).

CAPÍTULO 3. METODOLOGÍA

3.1 Tipo y diseño de la investigación

Para resolver el problema sobre la detección de fallos en compresores recíprocos de simple efecto doble etapa la investigación fue de enfoque **cuantitativo** con diseño **experimental** y de tipo **aplicada**, pues se manipularon las variables independientes en laboratorio a fin de observar sus efectos sobre la condición de la máquina en situaciones controladas de experimentación. En relación a lo expuesto, Hernández, Fernández, & Baptista, (2014) señala que la investigación de tipo cuantitativa tiene cuatro alcances: exploratoria, descriptiva, correlacional y explicativa. Para este autor, el alcance exploratorio se aplica para temas que necesitan información precisa de un contexto en particular, en tanto que el alcance descriptivo es útil para exponer con mayor detalle las características o dimensiones del tema de estudio, el alcance correlacional permite establecer el grado de asociación entre las variables y el alcance explicativo se orienta a indicar las causas de los eventos y fenómenos ocurridos en la investigación.

Partiendo de lo descrito, en la primera fase la investigación tuvo un alcance **exploratorio** que permitió conocer los últimos avances sobre la detección de fallos en compresores, centrándose en el modelado basado en datos a partir de las señales de vibración. Para el efecto, fue necesario indagar sobre los fallos más comunes en compresores, los procesos de adquisición de señales, técnicas de adquisición de señales y su pre-procesamiento. En la segunda fase, la investigación se enmarcó en un

alcance **descriptivo** que se utilizó para determinar cuáles son los fallos más comunes que se presentan en los compresores y cómo se manifiestan, y finalmente en la tercera fase el alcance de la investigación fue **explicativa**, pues se determinó cómo las señales de vibración inciden en la detección de fallos en compresores reciprocantes de simple efecto doble etapa. Por otro lado las actividades más relevantes que se llevaron a cabo para el desarrollo de la investigación fueron las siguientes:

- Primero se inició con la búsqueda y selección bibliográfica sobre sistemas de diagnóstico de fallos en compresores que tomen como base la señal de vibración. A partir de esta información se pudo conocer los fallos más comunes en compresores y se supo cómo procesar las señales de vibración. A partir de esta información se diseñó la guía para una adecuada adquisición de señales de vibración y se determinó una metodología para su pre-procesamiento.
- Una vez adquiridas y pre-procesadas las señales de vibración, se procedió a la optimización bayesiana de los hiperparámetros para determinar los mejores conjuntos de hiperparámetros para el diseño del modelo neuronal LSTM.
- Finalmente el modelo generado, se evaluó para definir su exactitud en función del número de aciertos sobre la totalidad de la muestra representativa de datos que no fueron utilizados en la construcción del modelo. Además se comparó el mejor modelo generado con otros enfoques tradicionales.

3.2 Unidad de análisis

El interés se centra en los compresores reciprocantes de simple efecto doble etapa. Para el desarrollo de la investigación fue necesario delimitar un tipo específico de máquina que esté dentro de la clasificación antes mencionada. Para el efecto se utilizó el compresor alternativo EBG250 ubicado en el laboratorio de vibraciones

mecánicas de la Universidad Politécnica Salesiana sede Cuenca, Ecuador que se muestra en la Figura 23.



Figura 23. **Compresor recíprocante de simple efecto doble etapa EBG250.**

Tomado de laboratorio de vibraciones mecánica UPS Cuenca, Ecuador

3.3 Población de estudio

La población de estudio se refiere a los compresores recíprocantes de simple efecto doble etapa y dentro de este, la población viene determinada por todas las series de tiempo $x(t)$ con $t \in \{1 \dots T\}$ siendo T el valor mínimo de periodicidad del proceso, que se pueden obtener del compresor recíprocante bajo las K posibles condiciones de descarga y P posibles modos de fallo de la válvula del compresor. Nótese que T puede ser valor constante en el que se garantice al menos un periodo del proceso. Lo anterior, para señales de vibración adquiridas en compresores recíprocantes a una tasa de

adquisición de 50000 samples/s, se puede garantizar con un $T = 250000$. No obstante, tanto K como P son infinitos al tratarse de estados continuos de valores de descarga y de modos de fallo en la válvula, respectivamente. Por lo tanto, el tamaño de la población es infinito.

3.4 Tamaño de la muestra

Una muestra representativa de la población anteriormente descrita contiene la información del proceso bajo las siguientes condiciones de descarga: frecuencia de rotación constante del motor de 57.7 Hz, frecuencia de rotación del cigüeñal de 12.8 Hz, presión constante del tanque en 3 bar y cuatro modos de fallo en cada válvula. Lo anterior se consiguió discretizando tanto el intervalo de descarga como el modo de fallo en 10 segundos con intervalos de adquisición aleatorios entre 10 y 60 segundos.

Por la naturaleza del proceso se asume que los cambios que pueden existir en las series temporales de la señal de vibración en estos subintervalos son despreciables. En consecuencia, se capturaron 15 señales de vibración por cada posible modo de fallo con 17 condiciones de la máquina, obteniendo un tamaño de la muestra de 255 señales de vibración.

3.5 Selección de la muestra

La muestra representativa de las señales emitidas por el compresor reciprocante, se obtiene de la discretización de los intervalos de condición de carga y modos de fallo en la válvula. Se asume que los cambios existentes en los subintervalos tienen una afección despreciable en el comportamiento del proceso (cambios insignificantes en la señal de vibración medida). Los subintervalos están uniformemente distribuidos en su intervalo y el inicio de la adquisición de la señal de vibración en cada una de las

combinaciones de subintervalos está sujeto a una distribución de Bernoulli con una probabilidad de éxito (captura de la señal) de 0.5. Lo anterior elimina el sesgo que pueda existir en la muestra por periodicidad del proceso.

3.6 Técnicas de recolección de datos

Inicialmente la investigación partió de la consulta y análisis de fuentes secundarias a través de la revisión bibliográfica en revistas científicas como IEEE, Scopus, Science direct, y Web of Science sobre sistemas de diagnóstico de fallos en compresores que tenían como parámetro de análisis la vibración. Luego la investigación fue de carácter experimental, pues la recolección de datos se dio en fuentes primarias a través de los equipos de medición (acelerómetros) y el software LabVIEW para la adquisición de las señales, la técnica aplicada fue la observación directa. Para la generación y validación del modelo, la investigación también fue de carácter experimental porque se tomaron los datos adquiridos para la optimización de los parámetros de pre-procesamiento e hiperparámetros de la red neuronal para finalmente identificar los mejores modelos generados y comparar con otros enfoques tradicionales. De esta forma se pudo generalizar el modelo propuesto para la detección de fallos en el compresor recíprocante.

3.7 Análisis de la información

Para el análisis de las señales de vibración adquiridas, se utilizó el software LabVIEW para mostrar el comportamiento de la aceleración en función del tiempo. Para el pre-procesamiento de la señal se utilizaron códigos programados en el software Python3 donde se establecieron parámetros de pre-procesamiento para disminuir la

resolución de cada señal y generar un incremento en el tamaño de datos. En la construcción del modelo se aplicó la metodología de redes neuronales recurrentes LSTM donde también se aplicó Python3. Finalmente para la validación del modelo se utilizó la técnica estadística holdout descrita en la sección 2.3.7; utilizando nuevamente el software libre Python3.

3.8 Hipótesis

3.8.1 Hipótesis general

Es posible construir un modelo basado en datos a partir de señales de vibración para la detección de fallos en compresores recíprocos de simple efecto doble etapa.

3.8.2 Hipótesis específica

- a. El pre-procesamiento de las señales adquiridas, influye en la construcción de un modelo basado en datos para la detección de fallos de un compresor recíproco de simple efecto doble etapa.
- b. El proceso de optimización de hiperparámetros mejora la exactitud del modelo diseñado para la detección de fallos del compresor recíproco de simple efecto doble etapa.
- c. La evaluación del modelo generado a partir de señales de vibración es capaz de generalizar la detección de fallos en el compresor recíproco de simple efecto doble etapa.

3.9 Identificación de las variables

En la formulación de hipótesis se identificaron sus variables independientes y dependientes, que se detallan a continuación:

Cuadro 5. Identificación de las variables

<i>Hipótesis General</i>	<i>Variables</i>
Es posible construir un modelo basado en datos a partir de señales de vibración, para la detección de fallos en compresores recíprocos de simple efecto doble etapa.	VI: Señales de Vibración VD: Modelo basado en datos y detección de fallos
<i>Hipótesis Específicas</i>	<i>Variables</i>
a. El pre-procesamiento de las señales adquiridas, influye en la construcción de un modelo basado en datos para la detección de fallos de un compresor recíproco de simple efecto doble etapa.	VI: Pre-procesamiento (a1). VD: Construcción del modelo basado en datos (a2)
b. Se requiere un proceso de optimización de hiperparámetros para mejorar la exactitud del modelo diseñado para la detección de fallos del compresor recíproco de simple efecto doble etapa.	VI: Optimización de hiperparámetros (b1) VD: Exactitud del modelo diseñado (b2).
c. La evaluación del modelo generado a partir de señales de vibración es capaz de generalizar la detección de fallos en el compresor recíproco de simple efecto doble etapa.	VI: Evaluación del modelo generado a partir de señales de vibración (c1) VD: Generalización de la detección de fallos (c2)

Fuente: Propia

3.10 Operacionalización de las variables

Las operaciones necesarias para la medición de las variables de las hipótesis formuladas en la investigación se detallan en el Cuadro 6. En él se determina una definición conceptual que parte de la teoría y una definición operacional que se refiere al contexto en el que se desarrolla la investigación. Adicionalmente se establece la dimensión de cada variable que orienta el objeto de interés a ser medido y el indicador que determina cómo se va a medir.

3.11 Matriz de consistencia

En el Cuadro 7 se resume la planificación de la investigación que se ha llevado a efecto. Donde se puede evidenciar que se parte de la descripción de los problemas para el planteamiento de las hipótesis, luego se identificaron las variables, se establecieron sus indicadores y finalmente se detallan las técnicas o instrumentos que se utilizaron para la recolección de datos. Es importante señalar que esta matriz sirvió de base para la ejecución de todas las tareas efectuadas en el desarrollo de la investigación.

Cuadro 6. Operacionalización de las variables

Variable	Definición Conceptual	Definición Operacional	Dimensiones	Indicador
a1. VI: Pre-procesamiento (a1).	Proceso de ajuste de la señal de vibración a una curva normal o Gaussiana, donde el promedio y la mediana coinciden en el mismo punto.	Incremento de la cantidad de señales que son ajustadas a la distribución Gaussiana para la construcción del modelo neuronal	Número de señales pre-procesadas.	Parámetros de pre-procesamiento.
a2. VD: Construcción del modelo basado en datos (a2)	Utilización de métodos de reconocimiento de patrones para un mapeo de las características en condiciones normales y en condiciones de fallo del sistema.	Se refiere a la selección de los mejores parámetros de pre-procesamiento para la construcción de la red LSTM para la detección de fallos del compresor.	Parámetros de pre-procesamiento.	Exactitud del modelo.
b1. VI: Optimización de hiperparámetros	Ajuste de parámetros necesarios para la creación de un modelo para la detección de fallos	Proceso iterativo para la obtención de los mejores conjuntos de hiperparámetros para la generación de un modelo neuronal.	Hiperparámetros de la red neuronal	Mejores conjuntos de hiperparámetros.
b2. VD: Exactitud del modelo diseñado	Utilización de métodos de reconocimiento de patrones para un mapeo de las características en condiciones normales y en condiciones de fallo del sistema.	Se refiere a la selección de los mejores modelos LSTM para la detección de fallos del compresor.	Complejidad del modelo	Exactitud del modelo
c1. VI: Evaluación del modelo generado a partir de señales de vibración	Utilización de métodos de reconocimiento de patrones para un mapeo de las características en condiciones normales y en condiciones de fallo del sistema.	Se refiere al análisis del mejor modelo LSTM para la detección de fallos del compresor.	Complejidad del modelo	Número de hiperparámetros.
c2. VD: Generalización de la detección de fallos	Detección y examinación de síntomas y síndromes para determinar la naturaleza del fallo y sus características como: tipo, situación y extensión.	Es el número de aciertos que tiene el modelo en la tarea de detección de fallos.	Nº de aciertos/(totalidad de una muestra representativa de datos no utilizados en la construcción del modelo)	Exactitud del modelo

Fuente: Propia

Cuadro 7. Matriz de consistencia

PROBLEMA	OBJETIVO	HIPOTESIS	VARIABLES	INDICADORES		TÉCNICAS/INSTRUMENTOS DE RECOLECCIÓN DE DATOS
GENERAL ¿Es factible desarrollar un modelo basado en datos de señales de vibración para detección de fallos en un compresor recíprocante de simple efecto doble etapa?	GENERAL Desarrollar un modelo basado en datos de señales de vibración para la detección de fallos en compresores recíprocantes de simple efecto doble etapa.	GENERAL Es posible construir un modelo basado en datos a partir de señales de vibración, para la detección de fallos en compresores recíprocantes de simple efecto doble etapa.				
ESPECÍFICOS a. ¿Cómo influye el pre-procesamiento de las señales adquiridas mediante la guía de adquisición de señales de vibración, en la construcción de un modelo basado en datos para la detección de fallos en compresores recíprocantes de simple efecto doble etapa?	ESPECÍFICOS a. Determinar cómo influye la aplicación de una técnica de pre-procesamiento de las señales adquiridas en la construcción de un modelo basado en datos para la detección de fallos en un compresor recíprocante de simple efecto doble etapa.	ESPECÍFICAS a. El pre-procesamiento de las señales adquiridas, influye en la construcción de un modelo basado en datos para la detección de fallos de un compresor recíprocante de simple efecto doble etapa.	VI: Pre-procesamiento. VD: Construcción del modelo basado en datos.	INDICADORES VI Parámetros de pre-procesamiento.	INDICADORES VD Exactitud del modelo.	Observación directa: experimento
b. ¿Cómo se construye un modelo basado en datos a partir de la optimización de hiperparámetros para la detección de fallos en compresores recíprocantes de simple efecto doble etapa?	b. Diseñar un modelo basado en datos a partir de la optimización de hiperparámetros para la detección de fallos en compresores recíprocantes de simple efecto doble etapa.	b. El proceso de optimización de hiperparámetros mejora la exactitud del modelo diseñado para la detección de fallos del compresor recíprocante de simple efecto doble etapa.	VI: Optimización de hiperparámetros VD: Exactitud de modelo diseñado.	INDICADORES VI Mejores conjuntos de hiperparámetros	INDICADORES VD Exactitud del modelo	Enfoque basado en datos
c. ¿La evaluación del modelo generado a partir de señales de vibración, generaliza la detección de fallos en un compresor recíprocante de simple efecto doble etapa?	c. Evaluar la capacidad de generalización del modelo basado en datos a partir de la señal de vibración para la detección de fallos en compresores recíprocantes de simple efecto doble etapa.	c. La evaluación del modelo generado a partir de señales de vibración es capaz de generalizar la detección de fallos en el compresor recíprocante de simple efecto doble etapa.	VI: Modelo generado VD: Generalización de la detección de fallos	INDICADORES VI Número de hiperparámetros	INDICADORES VD Exactitud del modelo	Base de datos

Fuente: Propia

CAPÍTULO 4. MODELO DE DETECCIÓN DE FALLOS PROPUESTO.

El sistema de detección de fallos basado en datos con el enfoque de redes neuronales recurrentes sigue el esquema mostrado en la Figura 24, donde se pueden identificar cuatro etapas y se utiliza como datos de entrada las señales de vibración en el dominio tiempo, que provienen de las válvulas de ingreso y de descarga del compresor recíprocante de simple efecto doble etapa. A continuación se desarrolla cada paso de la metodología.

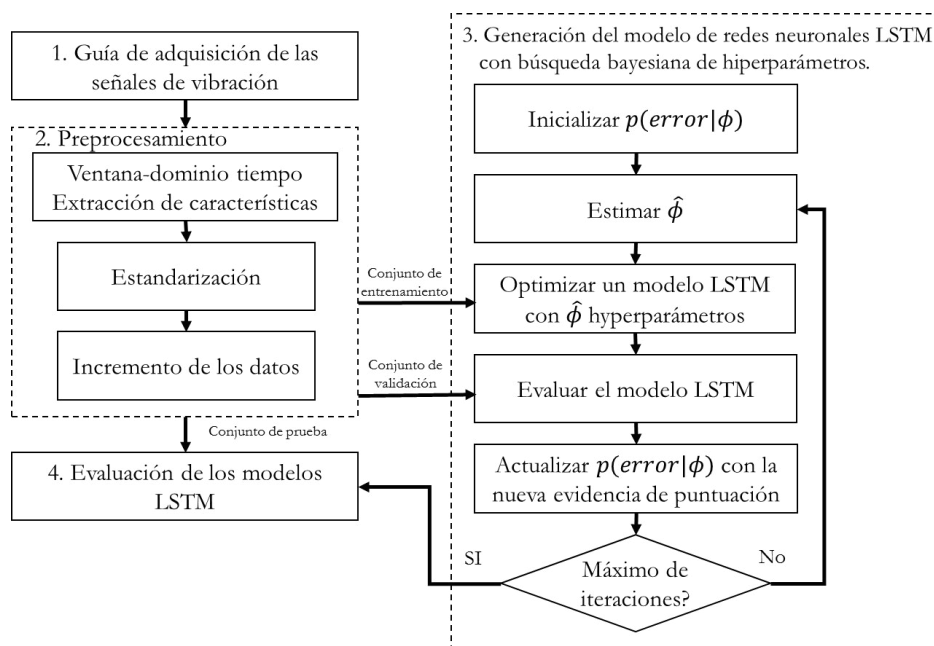


Figura 24. Metodología para la construcción de modelos LSTM a partir de señales de vibración para el diagnóstico de fallos de un compresor. Fuente: Propia

4.1 Guía de adquisición de las señales de vibración

4.1.1 Descripción del banco de experimentación

La parte experimental de la adquisición de datos se desarrolló en el laboratorio de vibraciones de la Universidad Politécnica Salesiana sede Cuenca, Ecuador. Para el efecto se utilizó el banco de experimentación que se muestra en la Figura 25.

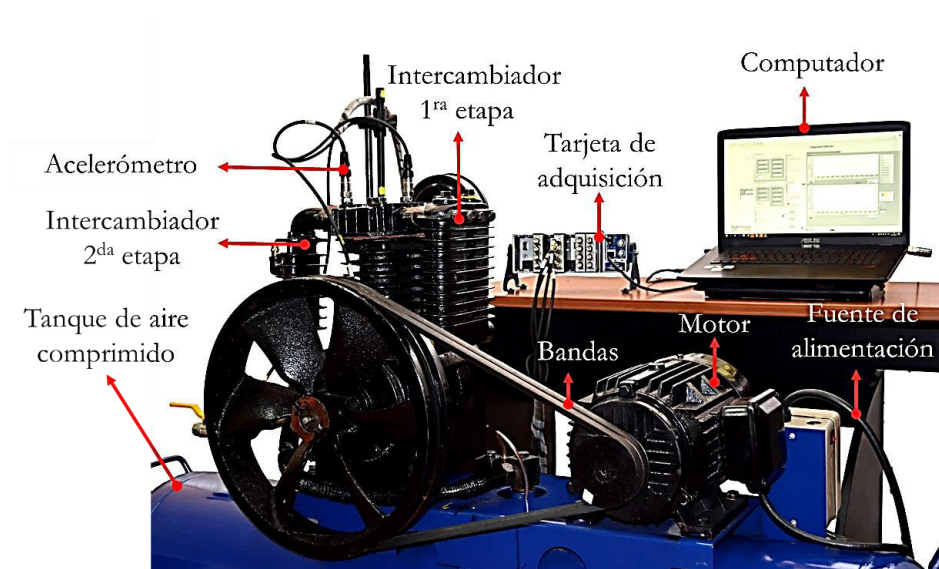


Figura 25. **Banco de experimentación.** Tomado del laboratorio de vibraciones mecánicas de la UPS, Cuenca Ecuador.

De acuerdo a la Figura 25 se puede constatar que el banco está compuesto de varios elementos, sin embargo para una mejor comprensión se agrupa en tres unidades principales que se ilustran en la Figura 26 y son las siguientes: (i) la unidad de mando que permite alimentar el compresor y el sistema de adquisición de datos, (ii) la unidad de adquisición de datos, que consta del sensor de aceleración (acelerómetros), las

tarjetas adquisición de datos de National Instruments y el computador portátil ASUS ROG GL752VW-DH74 y (iii) La unidad del compresor recíprocante EBG 250 en donde se simulaban los diferentes fallos a ser detectados.

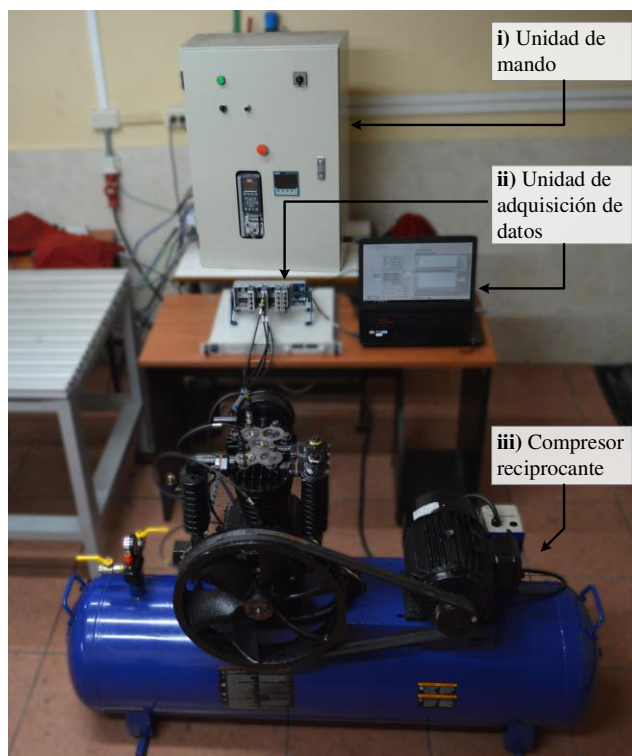


Figura 26. **Unidades del banco de experimentación.** Tomado del laboratorio de vibraciones de la UPS, Cuenca Ecuador.

4.1.2 *Sistema de adquisición de las señales de vibración*

En la Figura 27 se muestra el sistema de adquisición de datos. Las señales de vibración son censadas por un acelerómetro piezoeléctrico (b) modelo IPC 603C01 que se emplazó en el cabezote del compresor (a) modelo EBG 250; las características principales de los acelerómetros se indican en el Cuadro 8. La señal del acelerómetro se adquirió a 50 kS/s por la tarjeta (c) NI cDAQ-9234 cuyas características se indican en el Cuadro 9. A su vez, la tarjeta NI 9234 se conecta en el chasis (d) NI cDAQ-9188

que envían los datos vía Ethernet al computador portátil (e) ASUS ROG GL752VW-DH74 que dispone de LabVIEWTM y MatlabTM para la adquisición de las señales; las características del chasis NI cDAQ-9188 se indican en el Cuadro 10.

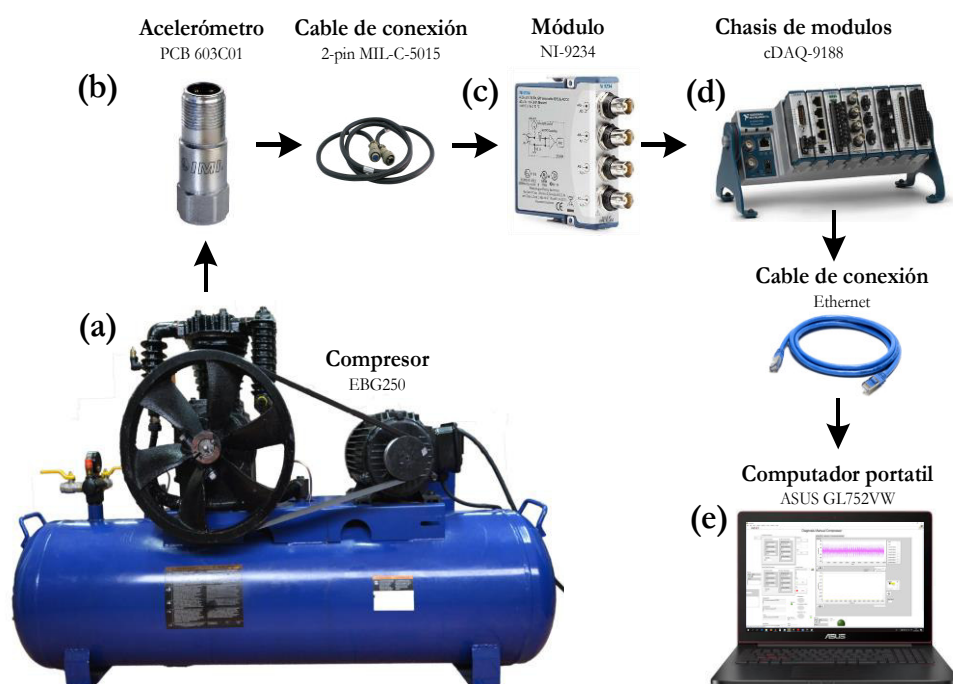


Figura 27. Esquema del sistema de adquisición de las señales de vibración.

Fuente: Propia.

Cuadro 8. Características del acelerómetro.

Marca y modelo	ICP - 603C01
Sensibilidad ($\pm 10\%$)	100 mV/g
Rango de medición	± 50 g
Rango de frecuencia	(± 5 dB) 0.5 a 10000 Hz
Frecuencia de resonancia	25 kHz
Resolución de ancho de banda (1 a 10000 Hz)	350 μ g

Fuente: National Instruments

Cuadro 9. Características de la tarjeta NI cDAQ-9234.

Máxima velocidad de muestreo	51.2 kS/s por canal
Rango de voltaje máximo	± 5 V DC
Número de canales	4
Resolución	24 bits; rango dinámico de 102 dB; filtros anti-aliasing
Acondicionamiento de señales	IEPE
Acoplamiento de AC/DC	Seleccionable por software

Fuente: National Instruments

Cuadro 10. Características del chasis NI cDAQ-9188.

Capacidad	Hasta 256 canales
Número de ranuras	8
Disparo integrado	Si
Potencia	15 W
Rango de entrada	9 - 30 V DC

Fuente: National Instruments.

También es importante especificar cómo está dispuesto el acelerómetro que tomará las señales de vibración que provienen de las válvulas de succión y de descarga. En la Figura 28 se ilustra la ubicación específica que se ha determinado para la ubicación del acelerómetro en la experimentación.

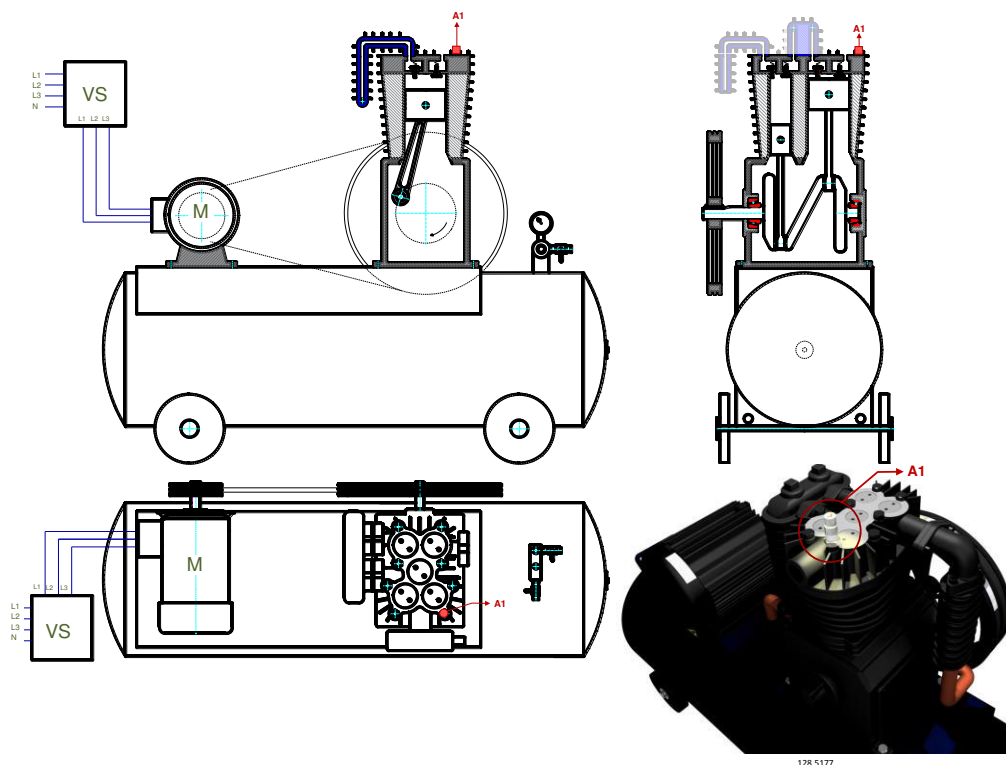


Figura 28. Ubicación del acelerómetro en el Compresor. Fuente: Propia

4.1.3 Configuración del banco de experimentación

a. Configuración del compresor

En el banco de experimentación se empleó un compresor recíprocante de doble etapa modelo EBG250 que está configurado por varios elementos que se ilustran en la Figura 29. Se muestra el motor que mediante las bandas transmite el movimiento a la polea-ventilador. Este a su vez mueve al cigüeñal y consecuencia de ello se mueven las bielas de la primera y segunda etapa del compresor. En la Figura 29b se presenta las válvulas de admisión y descarga tanto de la primera etapa como segunda etapa. Es necesario precisar que en estas válvulas se simularán los diferentes fallos que se describen más adelante y en el Cuadro 11 se presentan los principales datos técnicos del compresor.

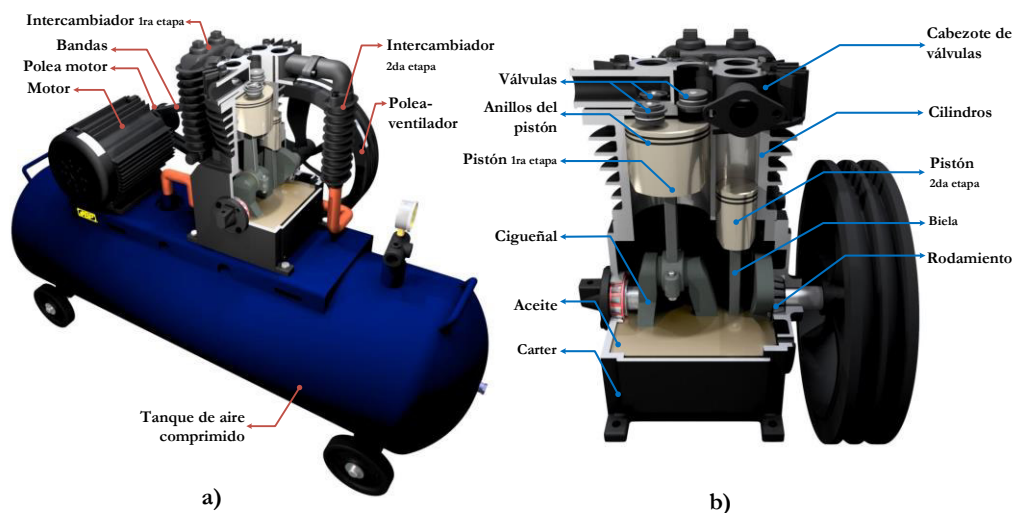


Figura 29. Configuración del compresor EBG250. Fuente: Propia.

Cuadro 11. Datos técnicos del compresor modelo EBG250

Modelo	EBG250
Marca	BP
Potencia	3.7 KW (5 HP)
Alimentación del motor	220 V CA
Velocidad del motor	3470 rpm
Tipo de transmisión	Dos bandas flexibles
Relación de reducción	4,5: 1
Nº de etapas	Dos etapas: una de baja y una de alta presión
Presión máxima	12 bar
Caudal aire	24.5 CFM
Posición del tanque	Tanque horizontal
Volumen del tanque	0.25 m ³
Tiempo de llenado	5 min
Altura total compresor	1.24 m
Largo total	1,79 m
Diámetro tanque	0,85 m
Tipo de lubricación	Baño de aceite
Aceite	SAE 20W50 (GT-1-Kendall)
Volumen de aceite en el deposito	3,2 litros
Tiempo para un ciclo completo.	0,078 segundos

Fuente: Propia

Adicionalmente, en la Figura 30 se presenta un esquema de la configuración del compresor y además se complementa la ilustración del funcionamiento. En la vista de detalle A, se puede apreciar claramente cómo funcionan las válvulas. El aire ingresa a la cámara de compresión por la válvula de admisión y el sistema comprime el aire que se descarga a través de la válvula de descarga. Este proceso se efectúa tanto en la primera como en la segunda etapa.

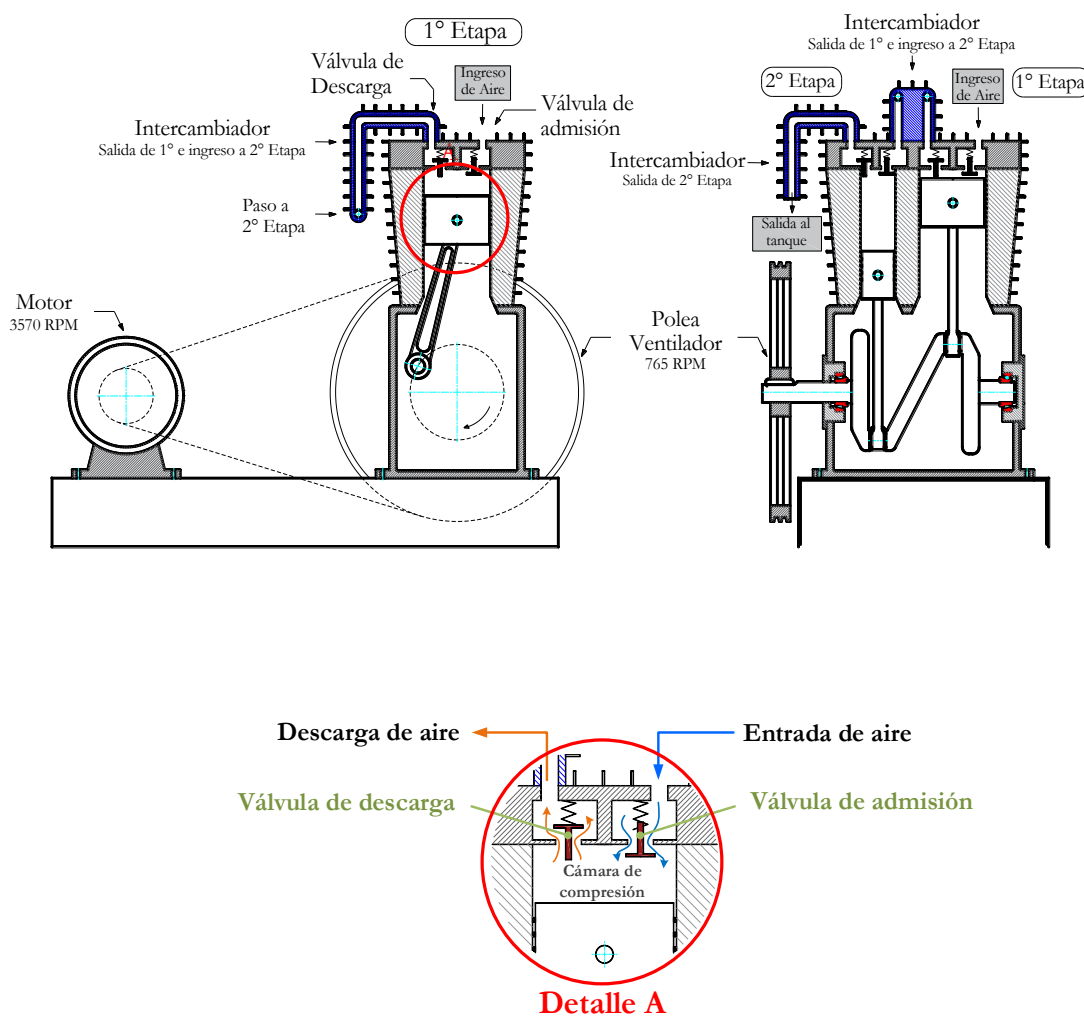


Figura 30. Esquema de funcionamiento del compresor EBG250. Fuente:

Propia

b. Configuración de los parámetros del banco

Para realizar la experimentación se deben controlar varios parámetros para que las pruebas puedan ser realizadas en las mismas condiciones. A continuación, se describen los diversos parámetros que se pueden regular.

1. La velocidad de rotación del motor se mantuvo a 3470 RPM para todas las pruebas
2. La presión del tanque se regula entre 2.9 y 3.1 bares, la regulación se realiza de forma manual y se verifica en el manómetro del tanque del compresor.
3. La temperatura de trabajo del compresor debe estar entre mínimo 38 °C y máximo 70 °C, el punto de control de la temperatura es la válvula de descarga de la segunda etapa, para esto se empleó un medidor láser.
4. Para la lubricación se utilizó el método de lubricación por baño, para todas las pruebas se mantuvo a la altura del 35% del depósito que asegura la lubricación de los rodamientos y pistones, se utilizó el aceite SAE 20W50 (GT-1-Kendall).
5. Se simulaban cuatro fallos, desgaste en el asiento de la válvula, corrosión en el plato de la válvula, fractura del plato de la válvula y rotura del resorte; todos los fallos se generaron artificialmente mediante proceso de mecanizado.

c. Configuración de las válvulas del compresor

En la Figura 31 se presenta un esquema del compresor en donde se presentan las válvulas del compresor de la primera y segunda etapa, todas las válvulas son de simple efecto y tipo placa; cumplen con la función de permitir la admisión y la descarga del aire en el cilindro tanto de la primera etapa como en la segunda etapa del compresor.

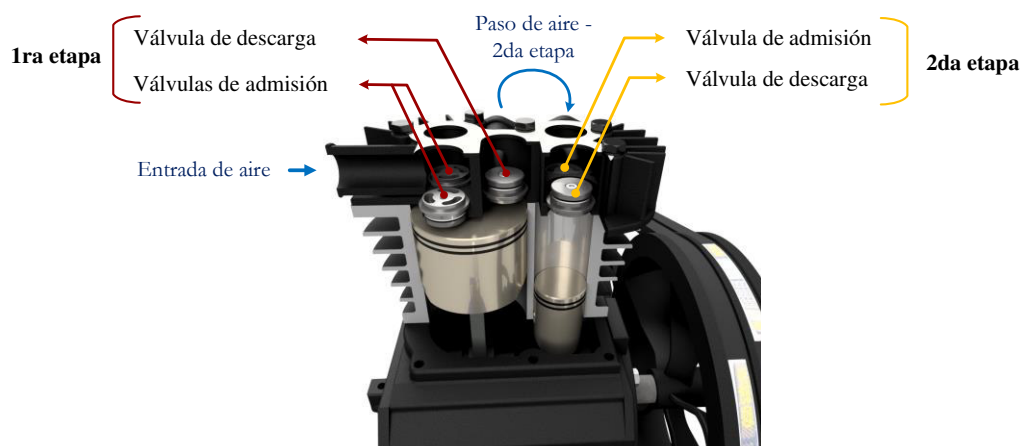


Figura 31. **Vista en corte del compresor.** Fuente: Propia.

El ensamble y sus partes se presentan en la Figura 32. Con base en esta figura, el Cuadro 12 presenta la descripción, función y características de las partes de la válvula. El estudio de los fallos y establecimiento de la base de datos se apoyó en el trabajo de Cajas Muñoz & Torres Díaz, (2018) estudiantes de la Universidad Politécnica Salesiana sede Cuenca, Ecuador, que forman parte del presente proyecto de investigación por medio de su proyecto de titulación.

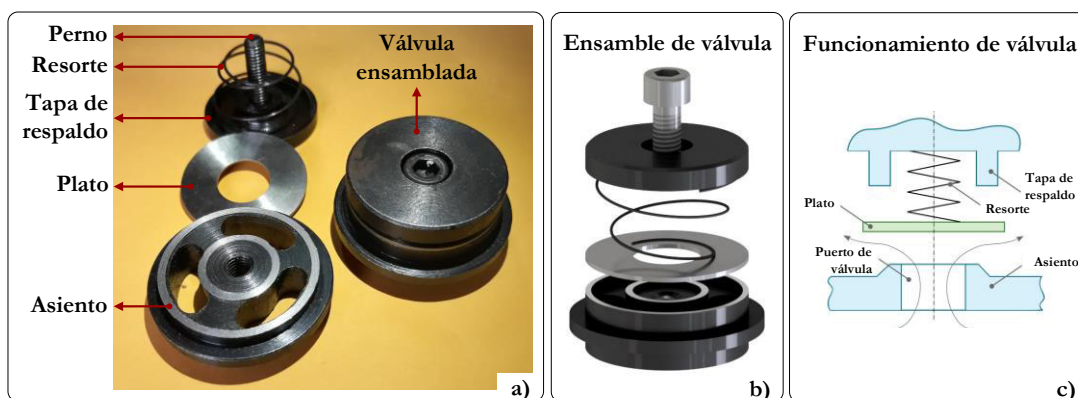


Figura 32. **Principio de funcionamiento de la válvula tipo placa de simple efecto.**

Fuente: Propia.

Cuadro 12. Descripción general de las partes principales de la válvula tipo placa.

Elemento	Descripción	Función	Características
Asiento	Es el cuerpo principal de la válvula y es de forma cilíndrica anular con perforaciones semicirculares a las cuales se les conoce como puerto de válvula.	Sirve de base para la válvula, está se asienta en el cabezal del compresor. Por su puerto de válvula fluye el aire. Hace sello con el plato de válvula bloqueando el ingreso de aire al cilindro.	Diámetro = 41,2 mm Altura = 12,5 mm Material: Acero al carbono
Tapa de respaldo	Es el cuerpo secundario de la válvula y tiene forma cilíndrica sólida.	Contiene los elementos de cierre: al plato de válvula y al resorte. Limita la apertura del plato de válvula.	Diámetro = 34,5 mm Altura = 5 mm Material: Acero al carbono
Plato	Es un disco de acero inoxidable martensítico que opera como elemento de sello con el asiento de válvula.	Abre o cierra el puerto de válvula. Cuando la válvula está cerrada, el plato de válvula se presiona contra el asiento de la válvula. Cuando está abierta se presiona contra la tapa de respaldo.	Diámetro = 34,5 mm Espesor=1,05 mm Material: Acero inoxidable
Resorte	Resorte divergente, de acero de alto contenido de carbono, tratado térmicamente y resistente a la fatiga	Sirve de medio para generar la fuerza que actúa sobre el plato de válvula para realizar el sello entre el plato y el asiento de válvula.	Resorte helicoidal del Diámetro alambre = 0,7 mm Material: Acero al carbono
Perno	Perno M6 de cabeza hexagonal de acero al carbono.	Sirve de medio de conexión entre el asiento de válvula y la tapa de respaldo. Permite calibrar la apertura del plato de válvula.	M6 x 40 mm Material: Acero al carbono

Fuente: Propia.

La válvula puede funcionar como una válvula de admisión o como una válvula de descarga, su funcionamiento depende de la orientación en la que se monte en el compresor. La Figura 33 muestra la forma de orientación necesaria para funcionar como válvulas de admisión o descarga.

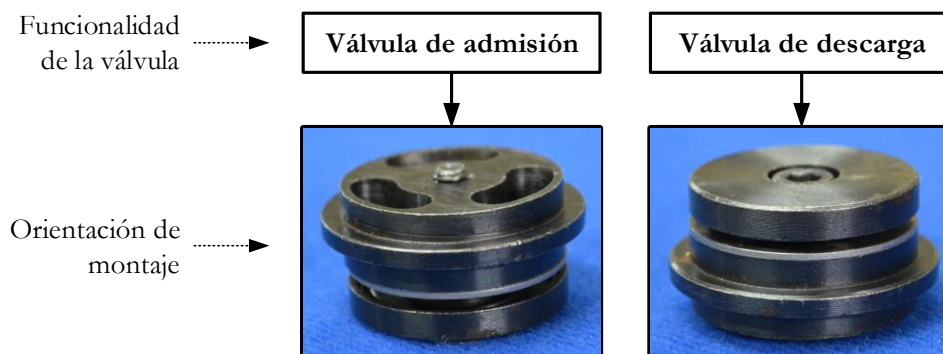


Figura 33. **Orientación de montaje de la válvula.** Fuente: Propia

4.1.4 Nomenclatura de los fallos en las válvulas del compresor

En la Figura 34 (a) se presenta un esquema de la vista superior del compresor y la ubicación de las seis válvulas. En la Figura 34 (b) se presenta la vista lateral en corte del cabezote del compresor que permite ubicar qué válvulas corresponden a la primera y segunda etapa. Para identificar las válvulas se utilizó la nomenclatura que se presenta en la Figura 35 con el apoyo de la vista superior que se presenta en la Figura 36.

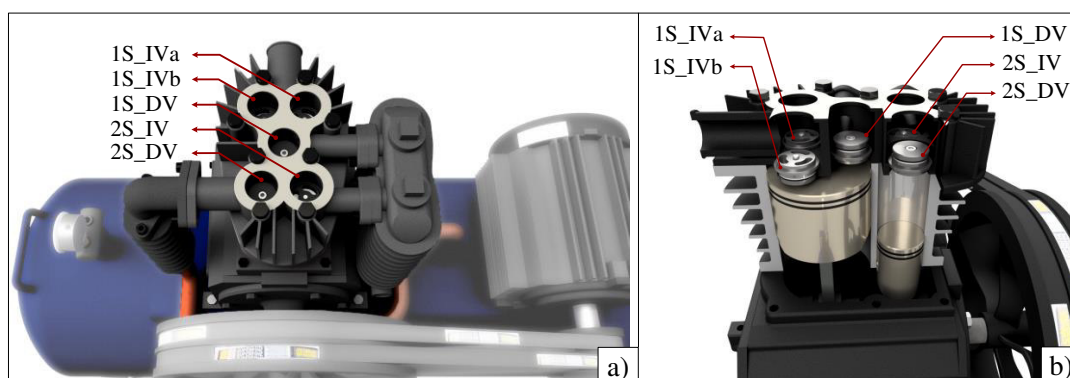


Figura 34. **Esquema 3D de la localización de las válvulas: a) vista superior y b) vista lateral en corte.** Fuente: Propia.

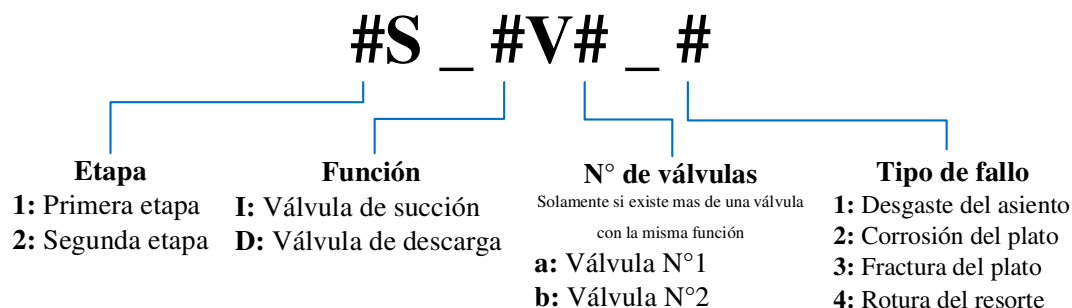


Figura 35. Codificación para la identificación de válvulas. Fuente: Propia.

El primer término de la nomenclatura “#S” corresponden a la posición de acuerdo con la etapa: (1S) para primera etapa y (2S) segunda etapa; el segundo término “#V”, corresponde a la función de la válvula: (IV) para válvula de admisión y (DV) para válvula de descarga; el tercer término que va luego del término “V#” solo se coloca cuando existen dos válvulas que cumplen la misma función y puede ser “a” o “b”; el último término “#V” corresponde al tipo de fallo generado y “_1” desgaste en el asiento de la válvula, “_2” corrosión del plato, “_3” fractura del plato y “_4” rotura del resorte. La Figura 36 ilustra lo descrito.

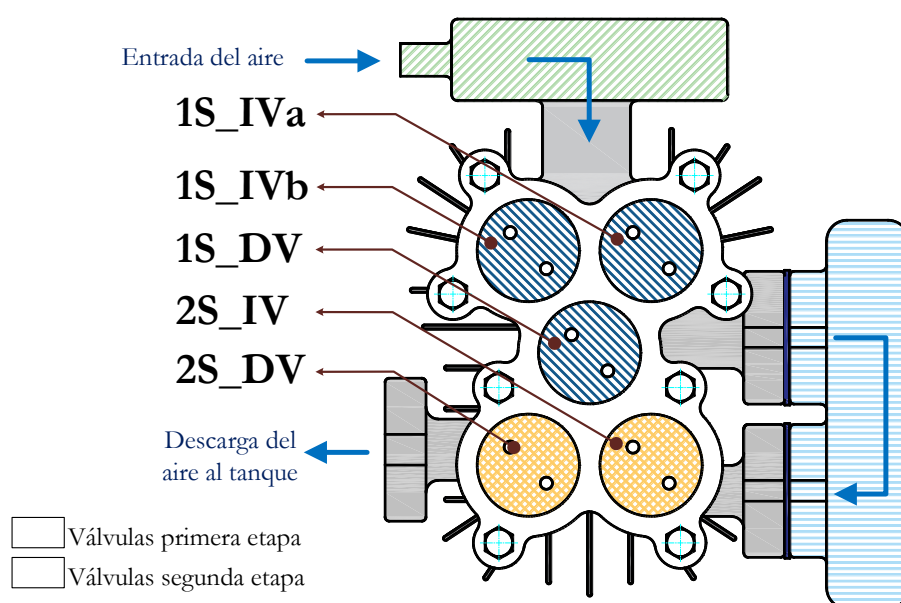


Figura 36. Vista superior del cabezote del compresor. Fuente: Propia.

4.1.5 Base de datos

- **Configuración de los fallos para la base de datos**

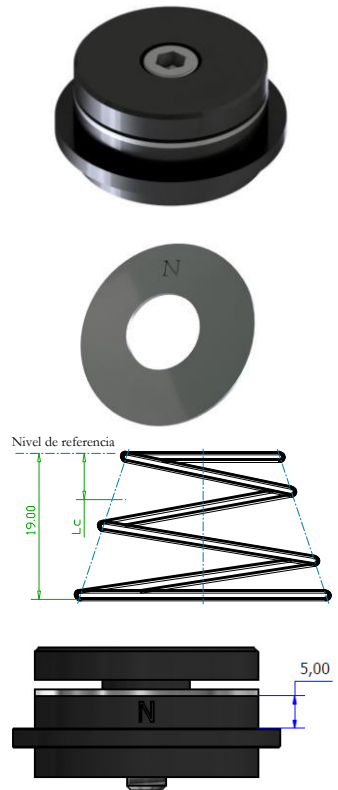

Se simularon 17 condiciones de trabajo en el compresor como se indica en el Cuadro 13, cada condición de trabajo se la denomina con la letra P. Se simuló la condición normal denominada P1, y 16 condiciones con diferentes fallos en diferentes válvulas (de P2 a P17). De esta forma, las señales de vibración fueron adquiridas de acuerdo a las cinco condiciones de la máquina: (1) condición saludable, (2) desgaste del asiento de la válvula, (3) corrosión de la placa de la válvula, (4) grieta de la placa de la válvula y (5) rotura de resorte. Los detalles de la construcción y ubicación de los fallos se presentan en el Cuadro 14.





Cuadro 13. Localización de los fallos en las válvulas del compresor.


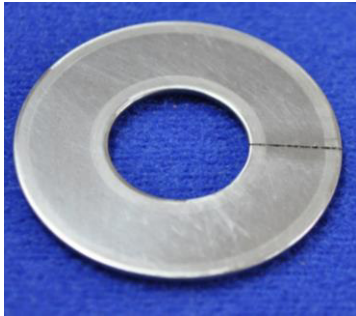
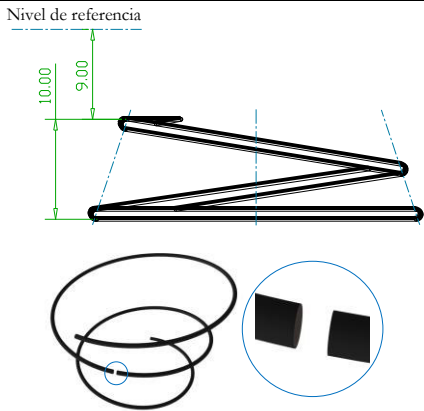

Condición	1S_IVa	1S_IVb	1S_DV	2S_IV	2S_DV
P1	N	N	N	N	N
P2	N	N	N	N	2S_DV_1
P3	N	N	N	N	2S_DV_2
P4	N	N	N	N	2S_DV_3
P5	N	N	N	N	2S_DV_4
P6	N	N	N	2S_IV_1	N
P7	N	N	N	2S_IV_2	N
P8	N	N	N	2S_IV_3	N
P9	N	N	N	2S_IV_4	N
P10	N	N	1S_DV_1	N	N
P11	N	N	1S_DV_2	N	N
P12	N	N	1S_DV_3	N	N
P13	N	N	1S_DV_4	N	N
P14	N	1S_IVb_1	N	N	N
P15	N	1S_IVb_2	N	N	N
P16	N	1S_IVb_3	N	N	N
P17	N	1S_IVb_4	N	N	N

Fuente: Propia

Cuadro 14. Detalles de la construcción de los fallos y su localización.

Código de la válvula y fallo	Tipo de fallo Descripción del fallo	Esquema del fallo	Fotografía del fallo
N	No fallo Condición normal	 <p>The schematic shows a 3D CAD model of a valve assembly. It includes a top cap, a central ring labeled 'N', a coiled spring with dimensions 19,00 and 5,00, and a bottom seat assembly.</p>	 <p>The photograph shows the physical components of the valve assembly, including the top cap, central ring, spring, and bottom seat assembly.</p>

Código de la válvula y fallo	Tipo de fallo Descripción del fallo	Esquema del fallo	Fotografía del fallo
1S_DV_1			
2S_IV_1	Desgaste en el asiento de la válvula		
2S_DV_1	Profundidad de desbaste = 1.44 mm		
1S_IVb_1			
1S_DV_2			
2S_IV_2	Corrosión en el plato de la válvula		
2S_DV_2	Diámetro= 2.50 mm		
1S_IVb_2			

Código de la válvula y fallo	Tipo de fallo Descripción del fallo	Esquema del fallo	Fotografía del fallo
1S_DV_3	Fractura del plato de la válvula		
2S_IV_3	Diámetro del hilo= 1.6 mm		
2S_DV_3	Longitud= total		
1S_IVb_3			
1S_DV_4	Rotura del resorte		
2S_IV_4	Longitud para el corte=9,00 mm		
2S_DV_4			
1S_IVb_4			

Fuente: Propia.

- **Bases de datos adquiridas**

Como se experimentó con 17 condiciones o clases desde P1 hasta P17, para cada clase P se repitió 15 veces la toma de datos y cada repetición tiene un peso de 3,7 MB; al tener 17 condiciones la base de datos tiene un peso total de 943.5 MB. Si se recuerda que los cuatro modos de fallo se implementaron en 17 posiciones diferentes, el Cuadro 15 muestra la codificación y la descripción de las condiciones de la máquina para las señales adquiridas y que son el insumo para el pre-procesamiento.

Cuadro 15. Configuración de fallos.

CÓDIGO DE FALLO	DESCRIPCIÓN
P1	Condición normal de todos los componentes
P2	Desgaste del asiento de la válvula de descarga en la segunda etapa.
P3	Corrosión de la válvula de descarga en la segunda etapa
P4	Fractura del plato de la válvula de descarga en segunda etapa.
P5	Rotura de resorte de la válvula de descarga en la segunda etapa.
P6	Desgaste del asiento de la válvula de succión en la segunda etapa.
P7	Corrosión del plato de la válvula de admisión en la segunda etapa.
P8	Fractura del plato de la válvula de succión en la segunda etapa.
P9	Rotura de resorte de la válvula de succión en la segunda etapa.
P10	Desgaste del asiento de la válvula de descarga en la primera etapa.
P11	Corrosión del plato de la válvula de descarga en la primera etapa.
P12	Fractura del plato de la válvula de descarga en la primera etapa.
P13	Rotura del resorte de la válvula de descarga en la primera etapa.
P14	Desgaste del asiento de la válvula de succión N°2 en la primera etapa
P15	Corrosión de plato de la válvula de succión N°2 en la primera etapa
P16	Fractura del plato de la válvula de succión N°2 en la primera etapa.
P17	Rotura de resorte de la válvula de succión N°2 en la primera etapa.

Fuente: Propia.

4.1.6 Software utilizado para la adquisición de datos

La adquisición de la señal se realizó en LabVIEW, donde se tiene el programa con la interfaz que se muestra en la Figura 37.

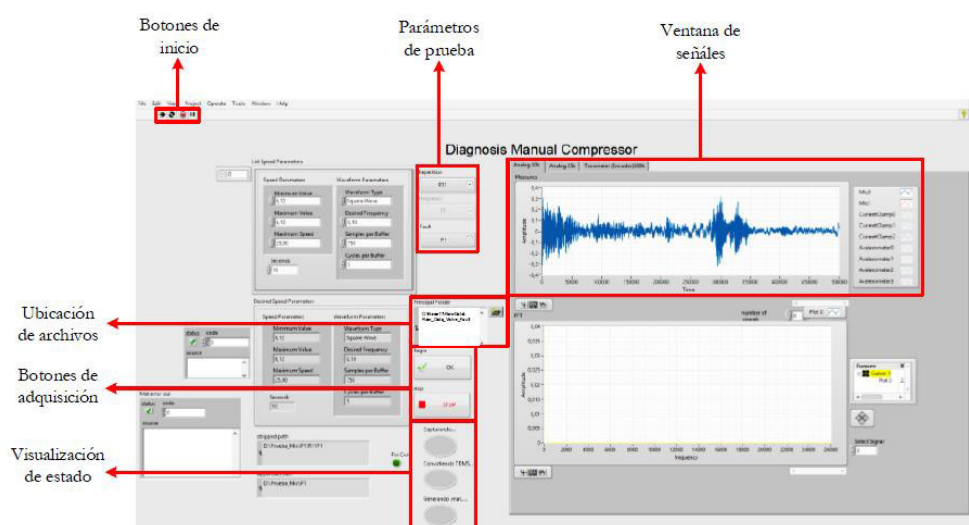


Figura 37. **Interfaz gráfica del programa de adquisición de señales.** Tomado del laboratorio de vibraciones mecánicas de la UPS Cuenca, Ecuador.

Como entrada se tiene las señales que se captan desde el compresor a través del acelerómetro y el resultado es la representación de la señal en el dominio del tiempo que tiene la forma ilustrada en la Figura 38. Se puede identificar los puntos referenciales a un ciclo de compresión. De esta forma el punto A señala el punto muerto superior, el punto B representa el cierre de la válvula de descarga, el punto C señala la apertura de la válvula de succión, D nuevamente señala un punto muerto pero inferior, E señala el cierre de la válvula de succión, en tanto que F determina la apertura de la válvula de descarga y el punto G muestra el punto muerto superior; todos estos puntos corresponden a la segunda etapa de compresión. Por otro lado se tienen los puntos que determinan la operación de la primera etapa de compresión, donde se

tienen: T que señala el punto muerto inferior, U que indica el cierre de la válvula de succión, V detalla la apertura de la válvula de descarga, W es el punto muerto superior, X indica el cierre de la válvula de descarga, Y denota la apertura de la válvula de succión y Z indica el punto muerto inferior. Cada punto que se describió se resume en el Cuadro 16.

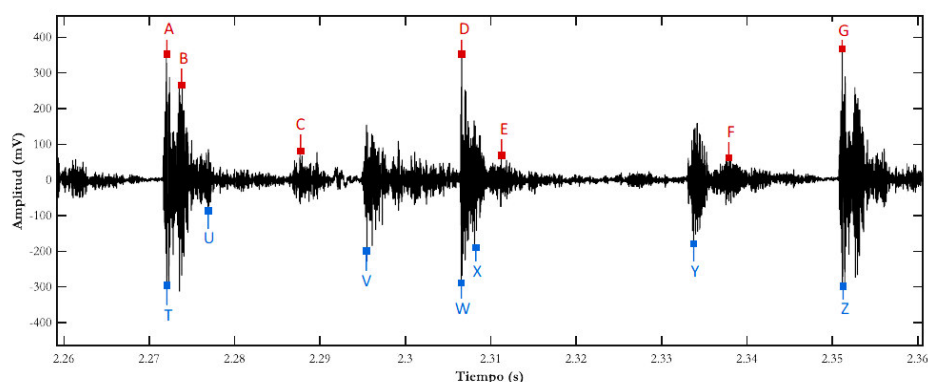


Figura 38. Señal de vibración de un ciclo de giro del cigüeñal en el dominio del tiempo del compresor EBG250 adquirida por el acelerómetro A1. Tomado del software LabVIEW.

Cuadro 16. Características de la señal de vibración de un ciclo de giro de cigüeñal en el dominio del tiempo del compresor EBG250, adquirida por el acelerómetro.

Segunda etapa		Primera etapa	
A	Punto muerto superior	T	Punto muerto inferior
B	Cierre de válvula de descarga	U	Cierre de la válvula de succión
C	Apertura de la válvula de succión	V	Apertura de válvula de descarga
D	Punto muerto inferior	W	Punto muerto superior
E	Cierre de la válvula de succión	X	Cierre de válvula de descarga
F	Apertura de la válvula de descarga	Y	Apertura de la válvula de succión
G	Punto muerto superior	Z	Punto muerto inferior

Fuente: Propia.

4.2 Pre-procesamiento

Como se vio en la sección anterior, el proceso de adquisición de señales involucra la recopilación y almacenamiento de información proveniente de elementos específicos que son de interés del investigador como la elección del tipo de sensor, su ubicación, cantidad, el hardware de adquisición y almacenamiento. La base de datos adquirida contiene una cantidad de 15 señales por cada modo de fallo tanto en las válvulas de succión de la primera y segunda etapa de compresión como en las válvulas de succión y de descarga de la segunda etapa. La Figura 39, presenta muestras de señales de vibración en el dominio tiempo correspondientes al fallo por desgaste del asiento de la válvula en diferentes posiciones. Como es evidente, existe gran similitud en la amplitud y forma de las señales en condición normal (a) y con fallo (b), (c), (d) y (e); siendo muy difícil identificar patrones de comportamiento para las diferentes condiciones de la máquina con una simple inspección visual de la señal, por lo tanto es necesario un análisis alternativo basado en machine learning.

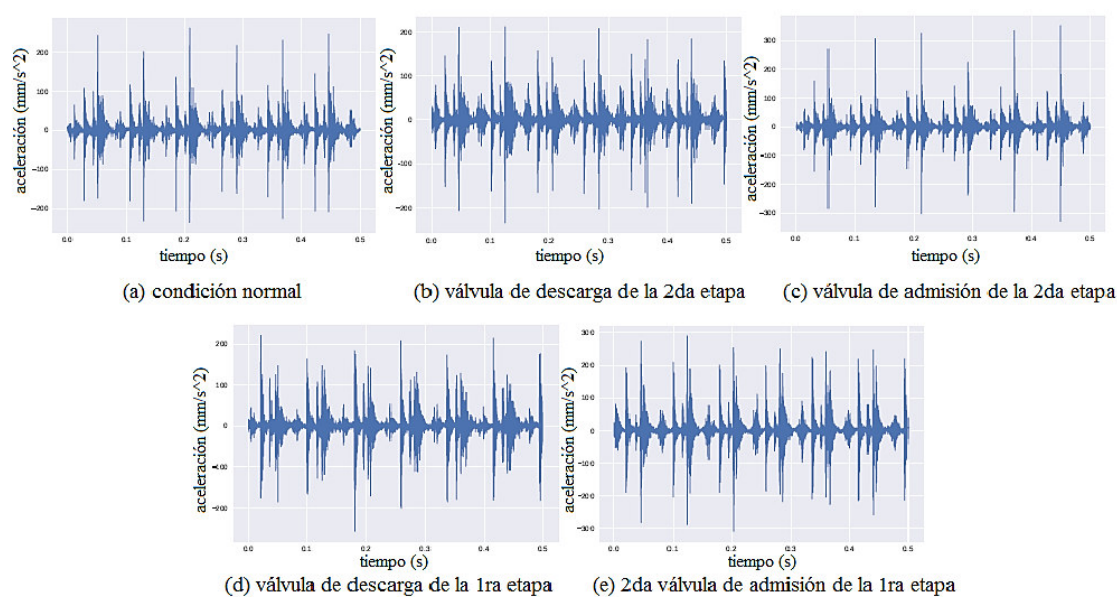


Figura 39. Señales de vibración correspondientes al desgaste del asiento de diferentes válvulas. Tomado de Python3

Las señales mostradas en la Figura 39 son un insumo para el modelamiento basado en datos, siendo importante recordar que para ello es necesario contar con una cantidad suficiente para generar un modelo. Con este antecedente, contar con una cantidad representativa de señales hace que el procedimiento de adquisición pueda volverse largo y costoso, incluso en muchos casos inviable. Una alternativa es el pre-procesamiento de las señales que tiene dos finalidades: (i) obtener señales con menor resolución sin perder sus atributos estadísticos y (ii) incrementar la cantidad de señales. En la Figura 40 se esquematiza el método a seguir con cada uno de sus pasos que se detallan a continuación:

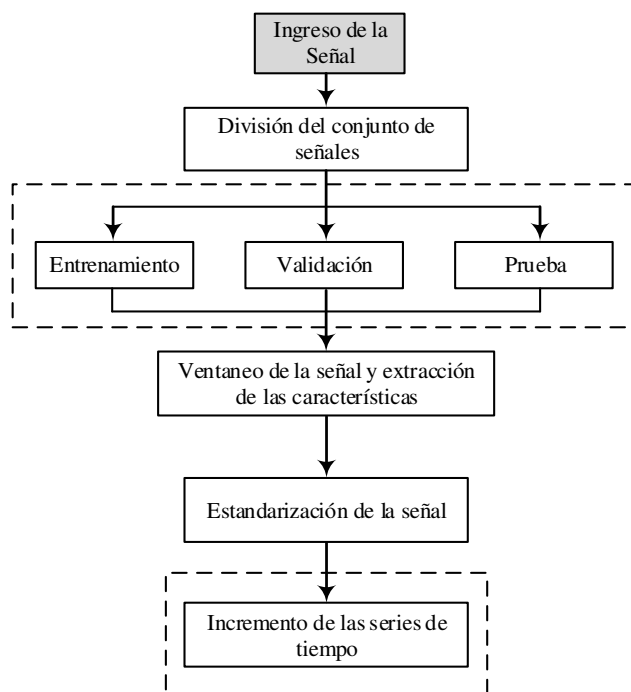


Figura 40. **Pre-procesamiento de las señales de vibración.** Fuente: Propia.

4.2.1 *Ingreso de la señal*

Como se mencionó anteriormente, las señales adquiridas desde el compresor pueden presentar perturbaciones o valores atípicos que posteriormente suelen interferir

en un correcto modelamiento. En tal virtud, luego de la etapa de adquisición, las señales son almacenadas en la base de datos del laboratorio de vibraciones de la Universidad Politécnica Salesiana de Cuenca, Ecuador para su pre-procesamiento, para lo cual es importante conocer las condiciones establecidas en la experimentación para la codificación de cada señal. De esta manera se han determinado 17 condiciones de la máquina con fallos generados en las válvulas de succión y de descarga, con 15 repeticiones tomadas para cada modo de fallo. En la Figura 41 se muestra la configuración de la base de datos.

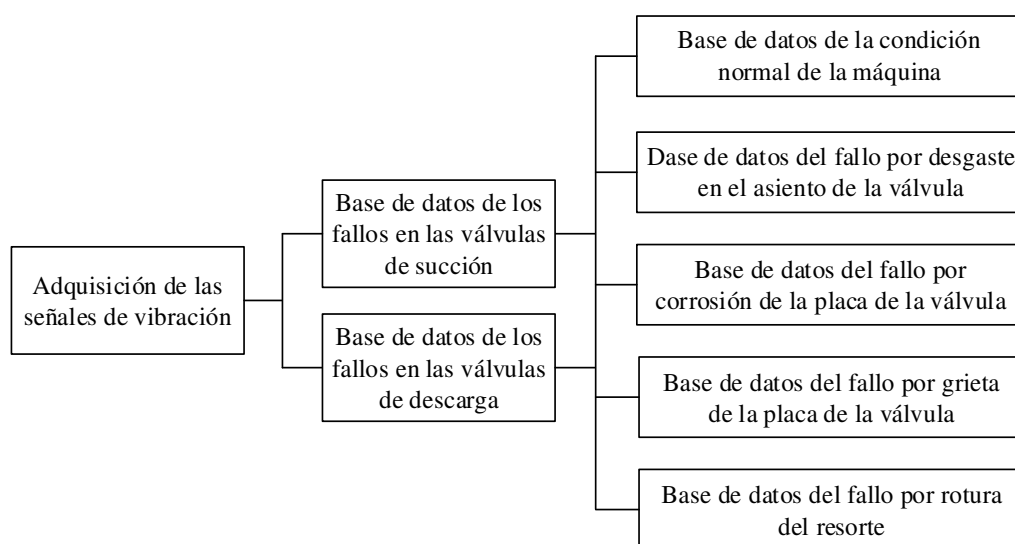


Figura 41. **Esquema de la base de datos.** Elaboración propia.

4.2.2 *División del conjunto de señales*

El siguiente paso es la división del conjunto de señales en tres subconjuntos denominados: entrenamiento, validación y prueba. Los subconjuntos de entrenamiento y validación son utilizados para el ajuste de los parámetros del modelo y el subconjunto de prueba se utiliza en una siguiente etapa para la evaluación del desempeño del modelo. Esta acción es necesaria para evitar un sesgo en la posterior evaluación del

modelo, pues dos de ellos se utilizan al momento de la generación de modelo y uno para su validación. Para el caso de estudio, la división se estableció de la siguiente forma: datos de entrenamiento con un 70% del total, datos de validación con el 15% y datos de prueba con el 15% restante. Este proceso se realizó de forma estratificada (de acuerdo a los modos de falla) y aleatoria para proporcionar una distribución homogénea de las señales, de tal forma que se obtengan subconjuntos con la cantidad adecuada y garantizando la presencia de todos los modos de fallo de la máquina en cada segmento. Considerando que inicialmente se obtuvo un total de 255 señales en el proceso de adquisición de datos, la división de los subconjuntos queda configurado de acuerdo a lo indicado en la Figura 42.

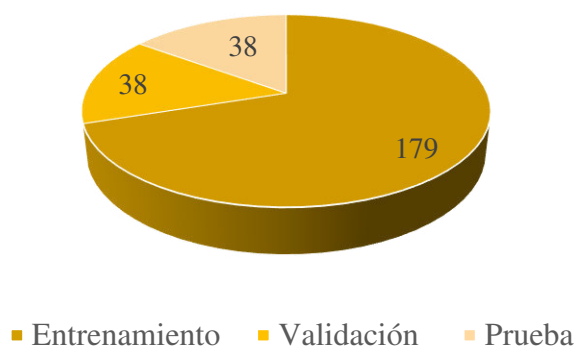


Figura 42. **División del conjunto de señales adquiridas.** Fuente: Propia.

4.2.3 Ventaneo de la señal y extracción de las características

Las series de tiempo de cada subconjunto son extensas pues están compuestas por una gran cantidad de samples (mediciones o puntos), por ejemplo para 10 segundos de adquisición a una $f_s=50\,000$ samples/s se obtiene una señal con 500 000 samples requiriendo un modelo recurrente altamente complejo para modelar la señal. Para abordar este problema se efectúa un “ventaneo” de la señal. Para este procedimiento se requiere considerar dos parámetros de ajuste: ventana de deslizamiento (v) y ventana

de atributos (w). Cada parámetro cumple una función específica que en conjunto permitirá disminuir la resolución de la señal y extraer sus características. De este modo, v determina cuánto se mueve w a lo largo de la señal y w especifica el ancho que se toma para la extracción de características. Es importante recordar que, si el ancho de v es menor que el ancho w , existirá superposición. En la Figura 43 se muestra el proceso descrito (ver sección 4.2.1).

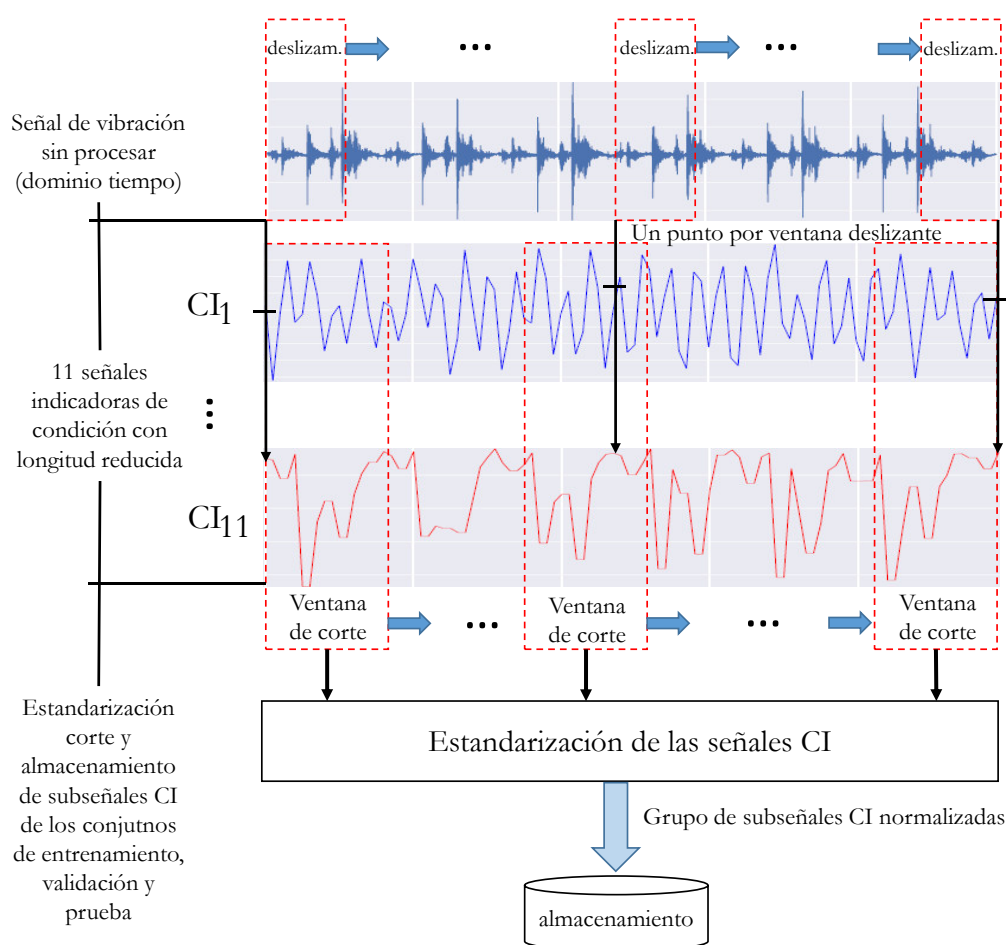


Figura 43. **Etapa de pre-procesamiento.** Elaboración propia

Dentro de cada ventana de atributos se procede a la extracción de indicadores de condición que permitan determinar el estado de la máquina. Esta acción favorece la reducción de la complejidad requerida por el modelo. La bibliografía consultada

(Sanchez, 2017) establece los siguientes indicadores estadísticos como útiles para el monitoreo de la condición en el dominio tiempo.

Valor medio	$CI_1(z) = \frac{1}{N} \sum z(i)$	[30]
-------------	-----------------------------------	------

Valor medio cuadrático	$CI_2(z) = \sqrt{\frac{1}{N} \sum z^2(i)}$	[31]
------------------------	--	------

Desviación estándar	$CI_3(z) = \sqrt{\frac{1}{N} \sum (z(i) - CI_1(z))^2}$	[32]
---------------------	--	------

Curtosis	$CI_4(z) = \frac{1}{N CI_3(z)} \sum (z(i) - CI_1(z))^4$	[33]
----------	---	------

Valor máximo	$CI_5(z) = \max(z)$	[34]
--------------	---------------------	------

Factor de cresta	$CI_6(z) = \frac{\max(z)}{CI_2(z)}$	[35]
------------------	-------------------------------------	------

Valor medio rectificado	$CI_7(z) = \frac{1}{N} \sum z(i) $	[36]
-------------------------	-------------------------------------	------

Factor de forma	$CI_8(z) = \frac{CI_2(z)}{CI_7(z)}$	[37]
-----------------	-------------------------------------	------

Factor de impulso	$CI_9(z) = \frac{\max(z)}{CI_7(z)}$	[38]
-------------------	-------------------------------------	------

Varianza	$CI_{10}(z) = CI_3^2(z)$	[39]
----------	--------------------------	------

Valor mínimo	$CI_{11}(z) = \min(z)$	[40]
--------------	------------------------	------

Donde N es el tamaño del vector z y el vector z contiene la subseñal entre el límite inferior y superior de la ventana de corte. Luego de la extracción de los indicadores $CI_1, CI_2, \dots, CI_{11}$, la señal tiene menor resolución, pues cada punto se representa en once dimensiones.

4.2.4 Estandarización de la señal

Cada señal obtenida en la etapa anterior se estandariza para evitar la saturación de las unidades del modelo neuronal debido a las funciones no-lineales descritas en las ecuaciones 21, 22, 23, 24, y 25. Los parámetros de distribución empíricos se estiman a partir del conjunto de entrenamiento y se utilizan para estandarizar cada subseñal del resto de conjuntos, evitando el sesgo causado por el uso de la información a partir de las señales de prueba. El proceso consta de dos partes: (i) Cálculo de la media $\hat{\mu}_k$ y desviación estándar $\hat{\sigma}_k$ para cada k-ésimo indicador y (ii) la normalización de cada señal en los tres subconjuntos de entrenamiento, validación y prueba, a través de la siguiente relación:

$$Norm\ CI_k(i) = \frac{CI_k(i) - \hat{\mu}_k}{\hat{\sigma}_k} \quad [41]$$

4.2.5 Incremento de las series de tiempo (señales)

Es conocido que para el modelamiento basado en neuronas se requiere contar con una cantidad suficiente de señales, por lo tanto, el tamaño del conjunto de entrenamiento es fundamental para la creación del modelo. De esta exigencia surge un inconveniente en la etapa de adquisición de datos, pues se requiere tiempo y altos costos para efectuar la experimentación. Para abordar este problema, se propone un paso adicional para incrementar la cantidad de señales. Si se considera que el tiempo de adquisición de cada señal es de 10s y que el proceso de compresión se efectúa en 0.158s, la aplicación de una ventana de corte de longitud l a cada señal CI y con un desplazamiento de un paso maximizará el conjunto resultante de señales. Para comprender mejor este procedimiento se desarrolla una fórmula para el incremento del tamaño de los datos:

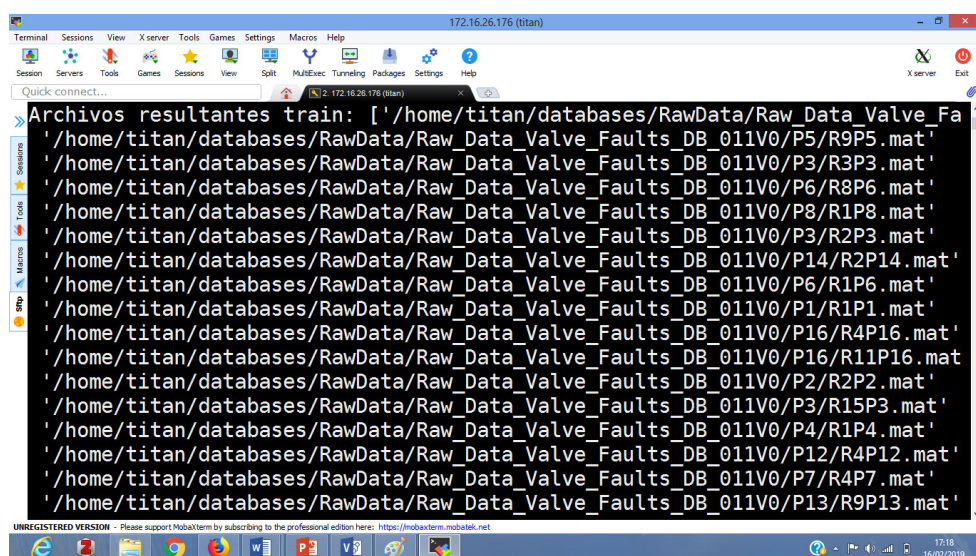
$$Tamaño\ de\ los\ Datos = int \left[\frac{(L - w)}{v} \right] - l \quad [42]$$

Donde L es la longitud total de la señal, w es la ventana de atributos, v es la ventana de deslizamiento y l es la longitud de la subseñal, es decir la ventana de corte.

4.2.6 Aplicación de software para el pre-procesamiento

Para llevar a la práctica la metodología del pre-procesamiento, se utilizó el software libre Python3. A continuación se detalla los pasos necesarios para los dos objetivos del pre-procesamiento: (i) disminuir la resolución de las señales originales y (ii) obtener un conjunto incrementado de señales estandarizadas.

1. El programa inicia con el ingreso para los archivos que contienen las señales adquiridas que están almacenados en la base de datos, Estos datos ingresan divididos en los tres subconjuntos de entrenamiento, validación y prueba. En la Figura 44 se ilustra una muestra de los archivos que se ingresa y corresponden al conjunto de entrenamiento.



```

Archivos resultantes train: ['/home/titan/databases/RawData/Raw_Data_Valve_Fa
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P5/R9P5.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P3/R3P3.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P6/R8P6.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P8/R1P8.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P3/R2P3.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P14/R2P14.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P6/R1P6.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P1/R1P1.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P16/R4P16.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P16/R11P16.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P2/R2P2.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P3/R15P3.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P4/R1P4.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P12/R4P12.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P7/R4P7.mat'
'/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P13/R9P13.mat'

```

Figura 44. **Archivos que contienen las señales adquiridas.** Tomado de la base de datos del laboratorio de vibraciones mecánicas de las UPS Cuenca, Ecuador.

2. Luego se extrae la información contenida en cada archivo, que corresponde a cada señal adquirida por el acelerómetro (referirse a la sección 4.1.3). Para este caso es una matriz que constan de 500 000 filas y una columna. Las filas representan los samples de la señal es decir su longitud y la columna indica que tiene una sola dimensión. En la Figura 45 se ilustra los datos que ingresan al pre-procesamiento.

The image shows a terminal window with a dark background and white text. The text displays several lines of numerical data representing signal samples. A red rectangular box highlights the line 'Tamaño: (500000, 1)', indicating the dimensions of the data matrix. The terminal window has a title bar and various icons in the top and bottom bars.

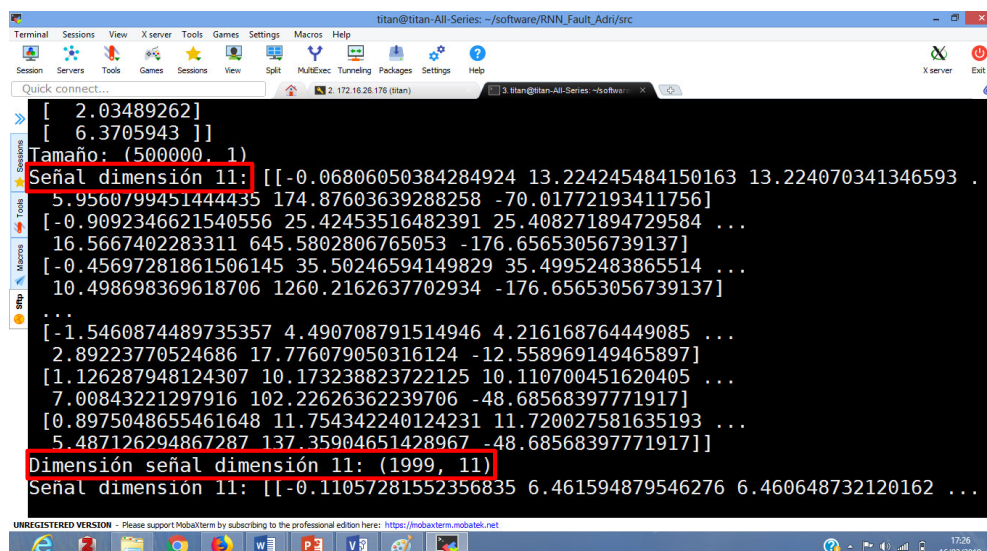
```

Señal a ser procesada: [[ 5.42371485]
 [ 7.0470052 ]
 [ 4.28147041]
 ...
 [-16.10662278]
 [-28.80409045]
 [-5.03028037]]
Tamaño: (500000, 1)
Señal a ser procesada: [[ -2.03399388]
 [-0.69973833]
 [-2.74370428]
 ...
 [-3.47743104]
 [-3.90205946]
 [12.89214411]]
Tamaño: (500000, 1)
Señal a ser procesada: [[ 5.2375132 ]
 [ 7.81660678]

```

Figura 45. Señal a ser pre-procesada. Tomado de Software Python.

3. A continuación se aplica una ventana deslizando para calcular 11 características estadísticas, como se muestra en la Figura 46. Donde se puede verificar que la señal tomada para ilustrar, ahora tiene una longitud de 1999 (de la señal original de 500000) con 11 dimensiones que corresponden a los 11 indicadores estadísticos que se detalló en la sección 4.2.3.



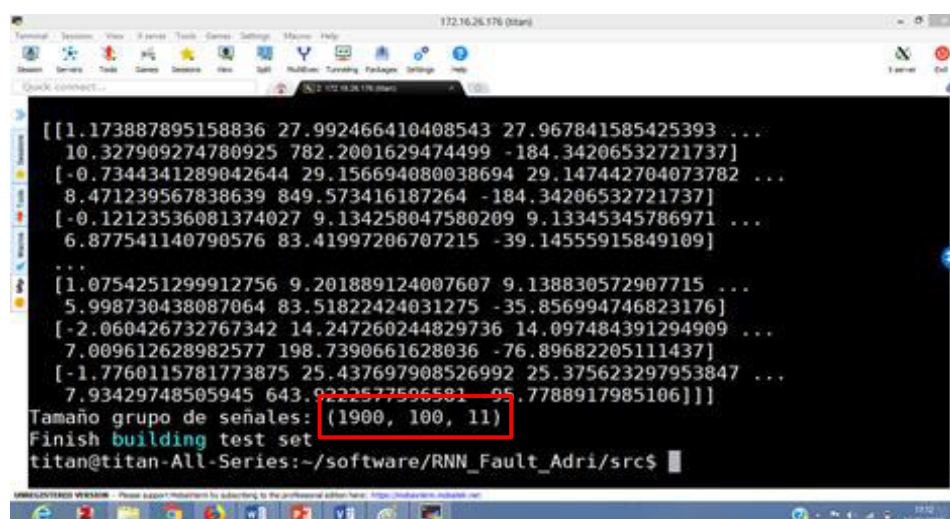
```

[ 2.03489262]
[ 6.3705943 ]]
Tamaño: (500000, 1)
Señal dimensión 11: [[-0.06806050384284924 13.224245484150163 13.224070341346593 .
5.9560799451444435 174.87603639288258 -70.01772193411756]
[-0.9092346621540556 25.42453516482391 25.408271894729584 ...
16.5667402283311 645.5802806765053 -176.65653056739137]
[-0.45697281861506145 35.50246594149829 35.49952483865514 ...
10.498698369618706 1260.2162637702934 -176.65653056739137]
...
[-1.5460874489735357 4.490708791514946 4.216168764449085 ...
2.89223770524686 17.776079050316124 -12.558969149465897]
[1.126287948124307 10.173238823722125 10.110700451620405 ...
7.00843221297916 102.22626362239706 -48.68568397771917]
[0.8975048655461648 11.754342240124231 11.720027581635193 ...
5.487126294867287 137.35904651428967 -48.68568397771917]]
Dimensión señal dimensión 11: (1999, 11)
Señal dimensión 11: [-0.11057281552356835 6.461594879546276 6.460648732120162 ...

```

Figura 46. Ventana deslizante y extracción de indicadores. Tomado de Python.

- Una vez que la señal se representa por 11 características o indicadores estadísticos se aplica una ventana de corte para obtener un grupo de subseñales a partir de una sola señal. En el ejemplo mostrado, de una señal original se obtienen 1900 subseñales con una longitud de 100 samples (mediciones) y con 11 dimensiones (indicadores). Los procedimientos de los pasos 1, 2 y 3 se realiza para las 255 señales que fueron adquiridas en la experimentación.



```

[[1.173887895158836 27.992466410408543 27.967841585425393 ...
10.327909274780925 782.2001629474499 -184.34206532721737]
[-0.7344341289042644 29.156694080038694 29.147442704073782 ...
8.471239567838639 849.573416187264 -184.34206532721737]
[-0.12123536081374027 9.134258047580209 9.13345345786971 ...
6.877541140790576 83.41997206707215 -39.14555915849109]
...
[1.0754251299912756 9.201889124007607 9.138830572907715 ...
5.998730438087064 83.51822424031275 -35.856994746823176]
[-2.060426732767342 14.247260244829736 14.097484391294909 ...
7.009612628982577 198.7390661628036 -76.89682205111437]
[-1.7760115781773875 25.437697908526992 25.375623297953847 ...
7.93429748505945 643.9223577506581 -95.7788917985106]]
Tamaño grupo de señales: (1900, 100, 11)
Finish building test set
titan@titan-All-Series:~/software/RNN_Fault_Adri/src$

```

Figura 47. Aplicación de una ventana de corte. Tomado de software Python

5. El último paso del pre-procesamiento es la estandarización. El resultado de este paso es la obtención de un grupo de señales del conjunto de entrenamiento ajustadas a una distribución normal, al igual que las señales del conjunto de validación y prueba. En la Figura 48 se muestra el resumen de las cantidades de señales pre-procesadas obtenidas en cada subconjunto.

```

[[0.5995993499114876 0.8532570454382918 0.8604078232774311 ...
 0.9872640013792613 0.5297842913303159 -1.4354557244328332]
[-0.3875570057935763 0.9468389956498034 0.9555338346794712 ...
 0.42157864866940364 0.6382105273278932 -1.4354557244328332]
[-0.0703552685614849 -0.6625872882126573 -0.6584447714244185 ...
 -0.06398529059610215 -0.5947881798421778 0.6403104920761461]
...
[0.5486655225962721 -0.65715102502112 -0.6580111472964578 ...
 -0.3317390760245018 -0.594630059032943 0.6873246461124798]
[-1.0734800366111792 -0.25159833000798015 -0.2581327882396694 ...
 -0.023746089136922038 -0.409200951822415 0.10060882699617094]
[-0.9263548533496415 0.6479018344416593 0.651364798078252 ...
 0.2579845153962675 0.30724861047526 -0.16933404913519323]]

tamaño test normalizado: (74100, 100, 11)
Conjunto de entrenamiento: (338200, 100, 11)
Conjunto de validación: (72200, 100, 11)
Conjunto de prueba: (74100, 100, 11)
para depuración:

```

Figura 48. Conjunto de señales estandarizadas. Tomado de Python.

4.2.7 Programación para el desarrollo de la metodología de pre-procesamiento.

A continuación se muestra la programación llevada a efecto en Python3 para la obtención de los resultados que se mostraron en la sección anterior. El diagrama de flujo de la Figura 49 orienta cómo se desarrolló la programación para el pre-procesamiento. El programa inicia con la creación de una ruta (Path) de entrada para los datos originales y se crea también una ruta (Path) de salida para los resultados. Luego se separa los datos en los subconjuntos de entrenamiento, validación y prueba, para posteriormente iterar y procesar los datos a través de la función “Parametric_generation”, y finalmente se guardan los datos. Este procedimiento se realiza para cada subconjunto.

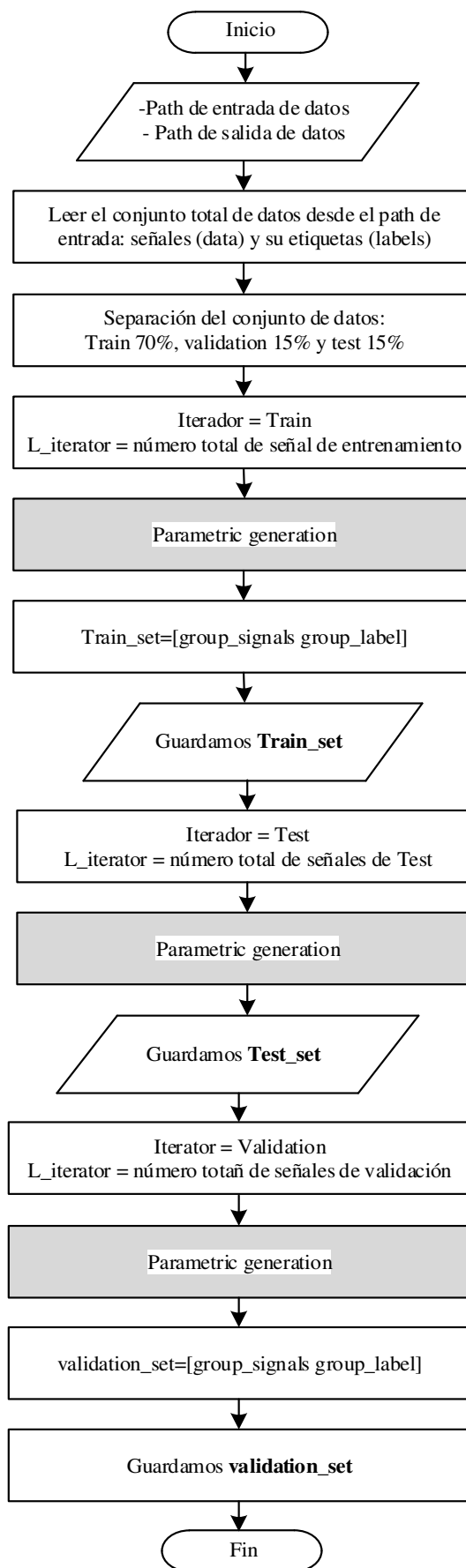


Figura 49. **Diagrama general del pre-procesamiento.** Fuente: Propia.

- Descripción de la función **Parametric_Generation**.

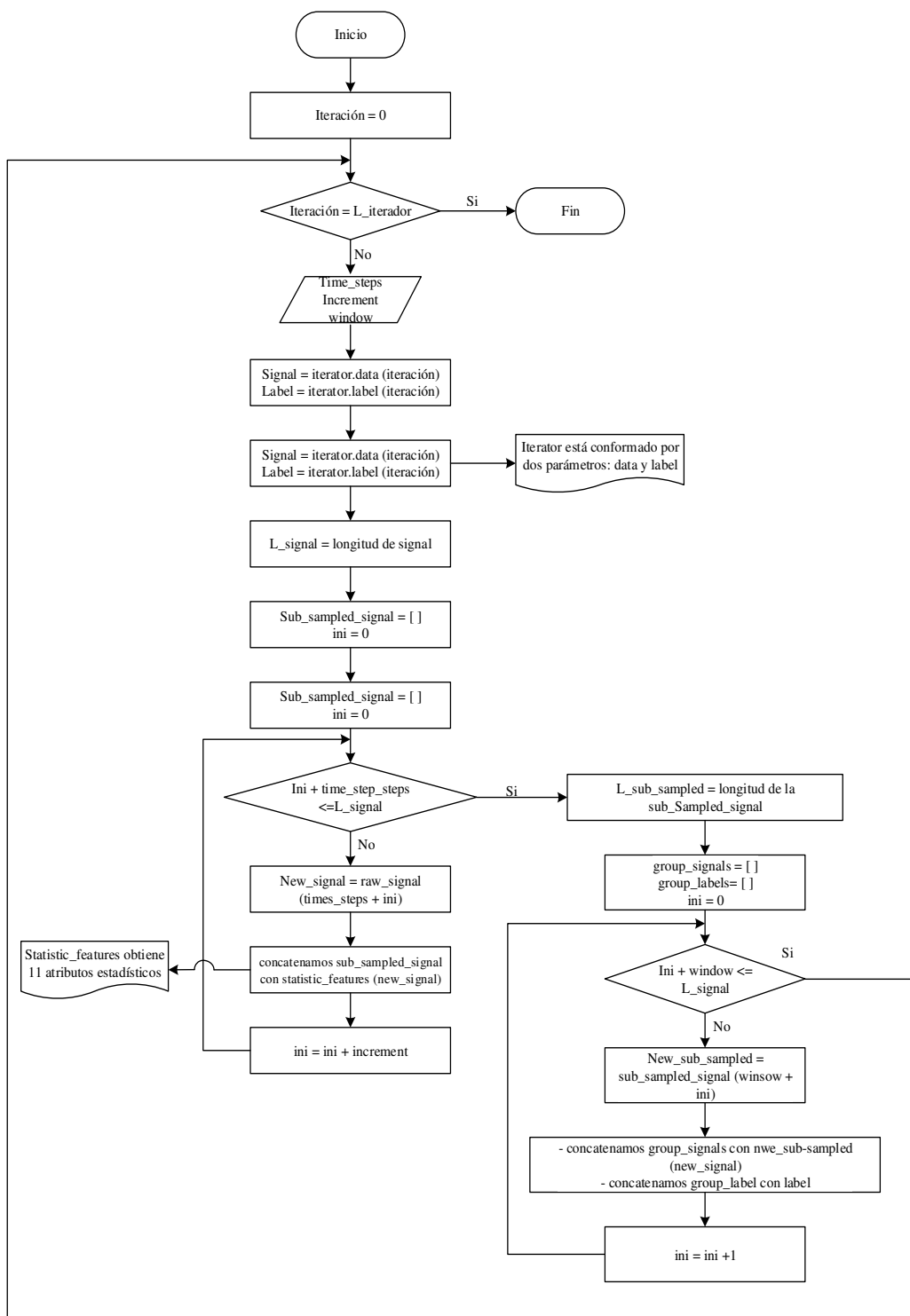


Figura 50. **Función Parametric_Generator**. Fuente: Propia.

4.3 Modelamiento con redes neuronales recurrentes LSTM y optimización bayesiana.

Finalizada la etapa de pre-procesamiento, la evaluación del estado de la máquina se realiza con el modelamiento basado en redes neuronales recurrentes Long Short Term Memory (LSTM). En esta etapa se utiliza el conjunto de señales previamente obtenido para la creación de un modelo LSTM con la búsqueda de los hiperparámetros que definan la red. Para esto se aplica el método de optimización bayesiana (OB), con la aplicación del algoritmo de Adam que se vio en la sección 2.3.11. Como último paso se evalúan las capacidades del proceso basado en optimización bayesiana y del modelo LSTM con las señales de vibración del conjunto de prueba bajo diferentes modos de fallo que pasaron a través de la etapa de pre-procesamiento.

4.3.1 Construcción de la red neuronal con búsqueda bayesiana de los hiperparámetros.

En esta sección se describe el paso 3 y 4 de la metodología mostrada en la Figura 24. Donde el paso 3 corresponde a la búsqueda bayesiana de hiperparámetros y en el paso 4 se realiza la prueba de los modelos LSTM. Luego del pre-procesamiento, el conjunto de entrenamiento obtenido $X = \{x_1, x_2, \dots, x_{N_{entren.}}\}$, está definido por la señal CI multidimensional contenida en un vector x_n para cada instante $i \in l$, como $x_n(i) = [CI_1(i), CI_2(i), \dots, CI_{11}(i)]_n$ y calculada para la n -ésima subseñal. Este subconjunto se utiliza para la creación del modelo \mathcal{M}_ϕ basado en neuronas a partir de los parámetros X y $\phi \in \mathcal{H}$.

El conjunto ϕ contiene los hiperparámetros que definen la arquitectura de la red, entre estos se tiene: la tasa de aprendizaje (lr), el número de capas (nl), el número de unidades (nu) y el tamaño de lote (b). El espacio de búsqueda \mathcal{H} contiene todos los posibles valores que pueden ser extraídos para la optimización del modelo. Además se

define la función objetivo $f(\mathcal{M}_\phi)$ que evalúa el error del diagnóstico del modelo. La optimización de la red se efectúa de acuerdo a la ecuación 26 que se abordó en la sección 2.3.11. En este caso la función objetivo que se optimiza es la puntuación del error del modelo, la expresión que lo define es:

$$error(P, \hat{P}) = 1 - \frac{1}{N_{valid}} \sum_{i=1}^{N_{valid}} 1(\hat{P}_i = P_i) \quad [43]$$

De esta ecuación, P_i es el modelo real y \hat{P}_i es el modelo estimado. El término $1(\cdot)$ es la función indicador del cumplimiento de la condición, si es 1 es verdadero, de contrario devuelve el valor de cero.

$$1(\cdot) = \begin{cases} 1 & \hat{P}_i = P_i \\ 0 & \hat{P}_i \neq P_i \end{cases} \quad [44]$$

Para el proceso de optimización de la red neuronal, se toma en consideración el proceso descrito en la sección 2.3.11. De acuerdo al cual, cada parámetro asocia una distribución previa relacionada con el conocimiento anterior del espacio de búsqueda. Para iniciar la optimización se toma valores empíricos que provienen de un tipo de distribución especificada por el investigador para cada hiperparámetro.

En el Cuadro 17 se resume lo mencionado, donde la distribución uniforme discreta se define para el intervalo e incremento fijados, esta distribución no da preferencia a algunos valores. Por otro lado la distribución logarítmica uniforme es una distribución que da preferencia a los valores cercanos al valor mínimo que típicamente ocurre en el hiperparámetro de la tasa de aprendizaje.

Inicialmente se parte del muestreo desde la distribución inicial. Luego el proceso continúa con el entrenamiento de la red LSTM a partir del conjunto de datos X con el algoritmo de Adam. A continuación, la red entrenada se evalúa en la función $f(\cdot)$ dada en la ecuación 26 con el conjunto de validación establecido. El valor obtenido en cada iteración se convierte en un nuevo valor de puntaje utilizado como fuente de información para obtener el siguiente conjunto de hiperparámetros. Este proceso

continúa durante 20 iteraciones determinadas de forma experimental como suficientes para alcanzar la convergencia del proceso de optimización. Además, se utiliza el $\hat{\phi}$ que minimiza la ecuación 26, es decir la función de estimación del error.

Cuadro 17. Distribuciones y límites que definen el espacio de búsqueda del hiperparámetro.

Hiperparámetro	Distribución	Mínimo valor	Máximo Valor	Incremento
<i>b</i>	Discreta uniforme	1000	20 000	1 000
<i>lr</i>	Logarítmica uniforme	Log(0.001)	Log(0.2)	-
<i>nl</i>	Discreta uniforme	1	3	1
<i>nu</i>	Discreta uniforme	5	30	5

Fuente: Propia.

Para terminar el proceso, el mejor modelo $\mathcal{M}_{\hat{\phi}}$ generado es evaluado en el subconjunto de prueba a través de la matriz de confusión y sus métricas derivadas, entre las que se tiene: verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos, exactitud, recall y precisión. Cada una de las métricas son evaluadas para el k-ésimo modo de fallo, de esta manera se tiene lo siguiente:

- Verdaderos positivos (TP_k): Porcentaje de muestras del k-ésimo modo de fallo correctamente identificadas como k-ésimo modo de fallo por el modelo.
- Falso negativo (FN_k): Porcentaje de muestras del k-ésimo modo de fallo que perteneciendo a un k-ésimo modo de fallo no fueron asignados a él.
- Falso positivo (FP_k): Porcentaje de muestras del k-ésimo modo de fallo identificado incorrectamente como k-ésimo modo de fallo por el modelo.

- Verdadero negativos (TN_k): Porcentaje de muestras del k-ésimo modo de fallo correctamente identificado por el modelo como no pertenecientes al k-ésimo modo de fallo.
- Precisión_k: capacidad del modelo para evitar clasificar incorrectamente muestras que no sean del k-ésimo modo de fallo como k-ésimo modo de fallo:

$$Precisión_k = \frac{TP_k}{TP_k + FP_k} \quad [45]$$

- Recall_k: capacidad del modelo para evitar clasificar incorrectamente muestras del k-ésimo modo de fallo como no k-ésimo modo de fallo:

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad [46]$$

- Fscore_k: Relación entre la precisión y el recall.

$$Fscore_k = 2 \frac{Precisión_k \text{ recall}_k}{Precisión_k + recall_k} \quad [47]$$

- Exactitud: Relación entre los éxitos y errores del modelo. Esta métrica se utiliza para medir el rendimiento general del modelo en K modos de fallo.

$$Exactitud = \frac{1}{K} \sum_{k=1}^K \frac{TP_k + TN_k}{TP_k + FP_k + TN_k + FN_k} \quad [48]$$

Con estas métricas se evaluará la capacidad de generalización del modelo LSTM. En el siguiente capítulo se muestran los resultados.

4.3.2 Aplicación de software para la creación del modelo

Al igual que en el pre-procesamiento, la aplicación de la metodología se efectuó en Python3. Cada señal de los tres subconjuntos obtenidos en la etapa anterior pasa a través del modelo LSTM según la arquitectura que se muestra en la Figura 51, según la cual se tiene como datos de entrada (capa de entrada) el conjunto de 11 indicadores o características estadísticas que caracterizan el comportamiento de la señal que tiene una menor longitud y dimensión de 11. Estas señales se utilizan para generar el modelo neuronal, validarlo en la optimización de hiperparámetros y probarlo en los cinco mejores modelos.

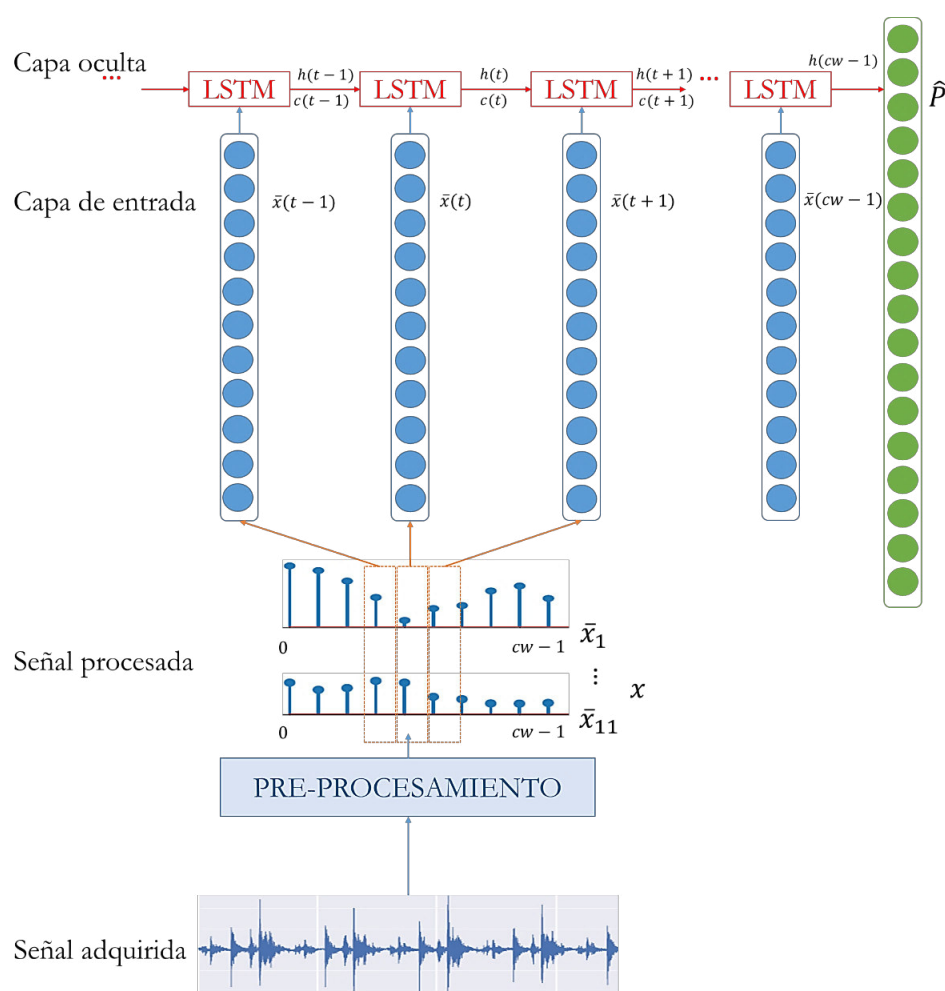


Figura 51. Representación de la red LSTM. Fuente: Propia.

Como se ilustra, a la red LSTM ingresan las 11 dimensiones en el instante $t-1$, luego ingresan 11 dimensiones en el instante t , de igual manera para el instante $t+1$ y así sucesivamente hasta el último instante. El retorno de las iteraciones de cada instante de tiempo es una predicción de cuál va a ser el fallo resultante, donde el número de neuronas de la capa de salida será igual al número de condiciones de la máquina establecidas en la etapa de adquisición de datos, que para este caso es de 17. Por lo tanto, en la neurona de salida donde aparezca un valor muy próximo a uno será la que se elija como la estimación correcta. De esta forma la salida estimada es comparada con la salida real y se calcula el error de cada iteración. Como se vio en la sección 2.3.11, cada vez que se efectúa una iteración la información resultante se agrega al campo de conocimiento para generar nueva información para obtener mejores resultados y ajustar los hiperparámetros de forma sucesiva hasta alcanzar un mínimo aceptable que permitirá obtener el menor error posible. Cuando esto sucede se puede concluir con el proceso de optimización de los hiperparámetros y el modelo se genera.

De acuerdo a lo que se explicó en el párrafo anterior, los datos que se ingresan a la red neuronal para la generación del modelo son las series temporales del conjunto de entrenamiento. Los valores que se indican en la Figura 52 corresponden a la señal de entrada, es decir, se ha tomado como ejemplo un conjunto de 10 000 subseñales de longitud 100 samples (mediciones) y de 11 dimensiones (indicadores estadísticos).

También es necesario ingresar los valores reales para que se pueda realizar una comparación entre la salida real (P) y la estimada (\hat{P}) con la finalidad de calcular el error del modelo. Estos datos se pueden verificar en el “batch” de series de ingreso que se muestra en la Figura 52 y de igual manera se determina el batch de los fallos reales. Nótese que debe haber congruencia entre el tamaño de batch de los fallos reales con el tamaño de tamaño de los datos de ingreso y para el ejemplo mostrado es de 10000. Pero a qué hace referencia el “batch”, pues es la cantidad de datos que se toma del conjunto de entrenamiento para cada iteración.

Con lo que se ha mencionado, se puede generar el modelo basado en redes neuronales recurrente LSTM, pues en esta sección se determinan qué hiperparámetros se consideran para configurar la arquitectura de la red.

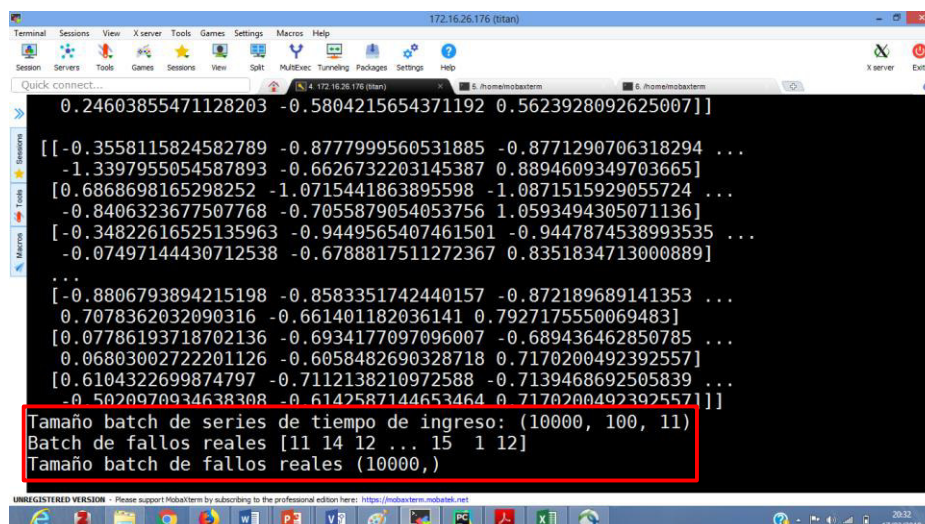


Figura 52. Datos de entrada a la red neuronal en su fase de entrenamiento.

Tomado de Python.

El resultado de las iteraciones que se efectúa con el algoritmo de Adam es la obtención de los cinco mejores conjuntos de hiperparámetros que definen el modelo neuronal LSTM. Esto se logra porque en la fase de validación se seleccionan los cinco mejores modelos, utilizando para el efecto los datos del conjunto de validación. A continuación se muestra un resultado de una iteración para un modelo LSTM-i, donde se muestra una exactitud de predicción de 0.646. Mientras mayor valor se alcance al final de las iteraciones, mejor predicción tiene el modelo. En la Figura 53 se muestra la exactitud del modelo tomado como ejemplo.

Por otro lado, los resultados de la arquitectura de la red neuronal se van almacenando en archivos de extensión .csv que pueden ser abiertos desde Excel. En la Figura 54 se ilustra los resultados del grupo de modelos que se van generando y almacenando. Se puede identificar la generación de la arquitectura para un grupo de los mejores modelos creados, donde se indica la tasa de aprendizaje, el tamaño del batch, el número de neuronas, el número de capas ocultas y la asertividad de la clasificación en la identificación de los fallos.

```

titan@titan-Ali-Series: ~/software/RNN_Fault_Adri/src
[[[-0.2834519567020713 1.3398558947213395 1.3502380880677713 ...
0.7994115300189591 1.1359460533959465 -2.6119055362431536]
[0.9366516196014677 1.6201252323287243 1.6281735105217652 ...
0.14667733812044906 1.5326962971068128 -2.6119055362431536]
[-0.08045077604002925 0.15103539894270163 0.15784703466910988 ...
-0.3428117456476963 -0.13231962891167975 0.06557403935008918]
...
[0.552635133389944 -0.07920919072701732 -0.07599356386066905 ...
-0.46648340605416266 -0.2985068625012662 0.3914708568439699]
[0.5245537850514934 -0.8160531644639637 -0.8182820613559652 ...
-0.4716955990954752 -0.6467333149438929 0.8433860536758028]
[-0.7047851389391632 -1.026872851207058 -1.0401190585469007 ...
-0.7414123379260211 -0.6978746186685143 1.0171420924892827]]]
Tamaño batch de series de tiempo de ingreso: (10000, 100, 11)
Batch de fallos reales [ 0 9 14 ... 11 13 3]
Tamaño batch de fallos reales (10000,)
At 196/ 200: COST 0.903/7.726 -- Acc 0.655/0.611
Exactitud validacion:
0.646124871001032

```

Figura 53. Exactitud del modelo generado. Tomado de Python.

	A	B	C	D	E	F	G	H	I	
1	loss,params,best_iter,train_time,cm									
2	0.8823529411764706,	{	'learning_rate': 0.0018202767778488784,	'batch_size': 18000,	'hidden_size': 2,	'num_layers': 3}	176			
3	0.34770983492211105,	{	'learning_rate': 0.12039740479867547,	'batch_size': 17000,	'hidden_size': 6,	'num_layers': 2}	196,8			
4	0.3587925288692553,	{	'learning_rate': 0.004445329156337353,	'batch_size': 20000,	'hidden_size': 16,	'num_layers': 2}	191,			
5	0.6832519569092459,	{	'learning_rate': 0.06155095270595364,	'batch_size': 17000,	'hidden_size': 2,	'num_layers': 2}	196,80			
6	0.6250484383476711,	{	'learning_rate': 0.004907799516732523,	'batch_size': 19000,	'hidden_size': 6,	'num_layers': 3}	186,1			
7	0.3479423389909323,	{	'learning_rate': 0.10109320542823043,	'batch_size': 14000,	'hidden_size': 10,	'num_layers': 2}	156,8			
8	0.36053630938541426,	{	'learning_rate': 0.012807150647649266,	'batch_size': 11000,	'hidden_size': 16,	'num_layers': 3}	171,			
9	0.364953886693017,	{	'learning_rate': 0.011692605047531612,	'batch_size': 2000,	'hidden_size': 10,	'num_layers': 2}	196,49			
10	0.15407269627218478,	{	'learning_rate': 0.09254155836206646,	'batch_size': 8000,	'hidden_size': 16,	'num_layers': 2}	156,6			
11	0.8797178950631636,	{	'learning_rate': 0.003059515544990586,	'batch_size': 16000,	'hidden_size': 2,	'num_layers': 1}	61,55			
12	0.8176005580097652,	{	'learning_rate': 0.003489414564386121,	'batch_size': 20000,	'hidden_size': 4,	'num_layers': 1}	196.6			

Figura 54. Resultado de la arquitectura neuronal LSTM. Tomado de los resultados de Python.

4.3.3 Programación para la creación del modelo

Para la generación del modelo se utilizó el software libre Python3. A continuación se indica el diagrama de flujo en (Figura 55). El programa inicia con la

definición de un espacio de búsqueda. Se procede a la búsqueda de hiperparámetros para la creación del modelo neuronal con 20 iteraciones para la optimización de los parámetros de pre-procesamiento y de 200 iteraciones en la optimización de los hiperparámetros de la red, dicho de otro modo, para la evaluación del modelo generado. Finalmente los resultados se almacenan en el path de salida.

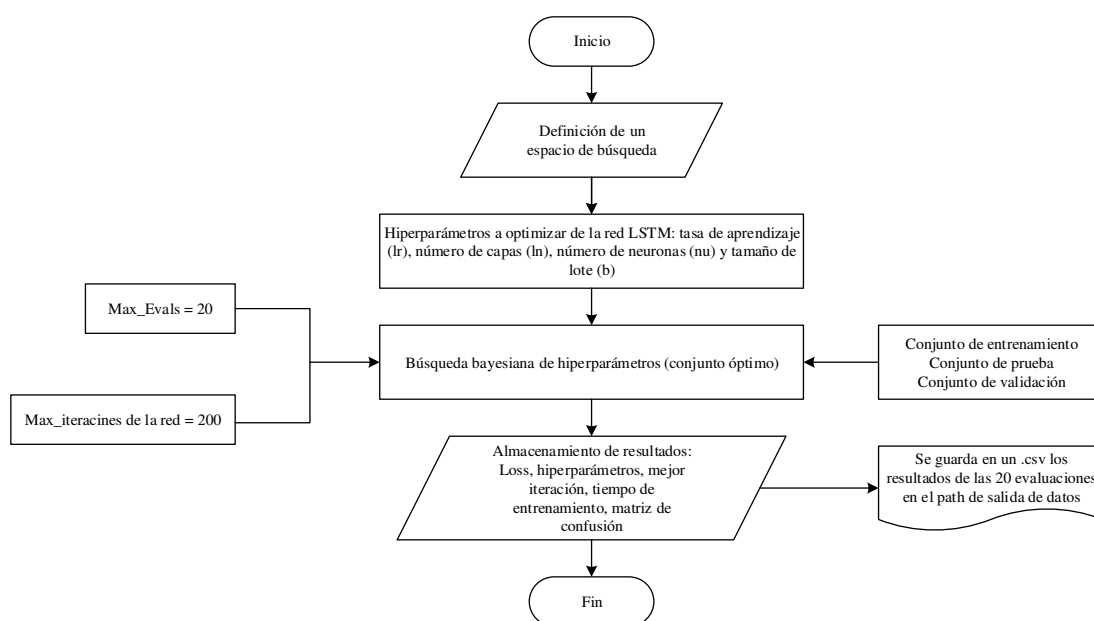


Figura 55. **Optimización de hiperparámetros.** Fuente: Propia.

4.3.4 Software utilizado para la generalización del modelo.

Una vez creados los modelos, se escogen los cinco mejores para realizar un análisis más profundo y de cada mejor modelo seleccionado se crean a su vez 10 modelos, con la finalidad de escoger al mejor modelo de entre los cinco mejores, como se muestra en la Figura 56.

```
titan@titan-All-Series:~/software/RNN_Fault_Adri/src$
titan@titan-All-Series:~/software/RNN_Fault_Adri/src$ python3 results.py
Identificacion de los mejores
  loss    ts    win    inc
0 0.081545 800.0 160.0 800.0
1 0.083986 600.0 180.0 300.0
2 0.086962 700.0 100.0 700.0
3 0.087286 500.0 120.0 250.0
4 0.089095 600.0 160.0 300.0
Grafica ts
Grafica subs length
Comparaciones
Resultados de mejores modelos: [[0.89959514 0.95832659 0.94261808 0.92618084 0.95039136 0.90531714
0.92928475 0.92761134 0.93144399 0.92693657]
[0.8561601 0.9127864 0.88725908 0.92864859 0.89686736 0.8982887
```

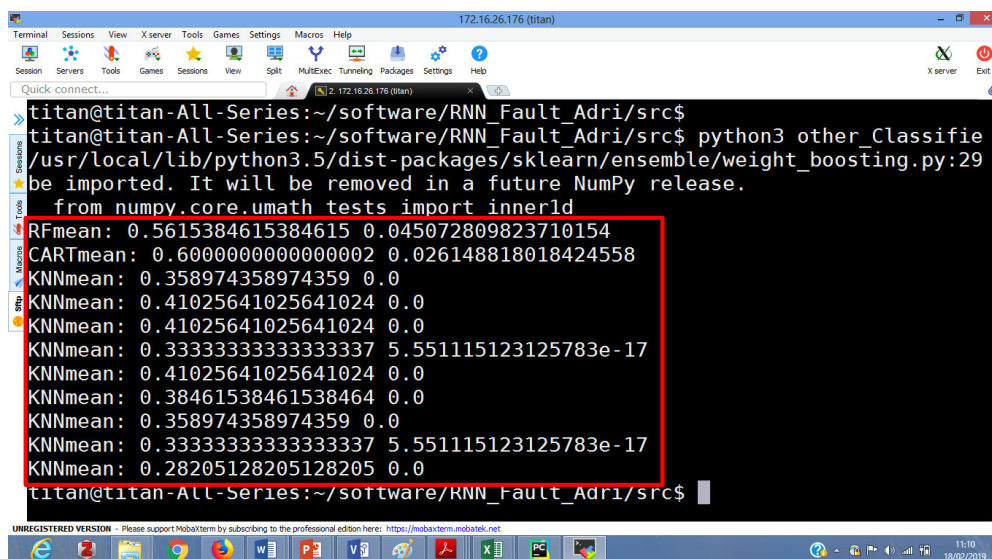
Figura 56. **Identificación de los cinco mejores modelos LSTM.** Tomado de Python.

Luego de la identificación del grupo de los cinco mejores modelos se procede a realizar una comparación entre ellos, a través de las pruebas estadísticas: Prueba de Kolmogorov-Smirnov y la prueba T-student para cada par de modelos. Estas pruebas ayudan a determinar la semejanza entre los cinco modelos para escoger finalmente al mejor modelo que servirá de pauta para evaluar su capacidad de generalización. Los resultados de este proceso se muestran en la Figura 57.

```
Resultados de mejores modelos: [[0.89959514 0.95832659 0.94261808 0.92618084 0.95039136 0.90531714
0.92928475 0.92761134 0.93144399 0.92693657]
[0.8561601 0.9127864 0.88725908 0.92864859 0.89686736 0.8982887
0.90977315 0.9204048 0.92967195 0.89720848]
[0.90103655 0.92144026 0.92165848 0.93805237 0.92572286 0.95889253
0.90324604 0.93857065 0.91022913 0.92089471]
[0.88563289 0.91842549 0.88580731 0.85400314 0.92749578 0.93017036
0.86475958 0.93674051 0.88883075 0.91592534]
[0.94259719 0.93132065 0.93267163 0.88265784 0.93101737 0.89820789
0.87791563 0.90609319 0.89707747 0.86936862]]
Tabla estadísticos
LSTM-1 & 0.57 & 0.907 & 0.025 & 0.869 & 0.943\\
LSTM-2 & 0.73 & 0.901 & 0.027 & 0.854 & 0.937\\
LSTM-3 & 0.96 & 0.924 & 0.017 & 0.901 & 0.959\\
LSTM-4 & 0.93 & 0.904 & 0.021 & 0.856 & 0.93\\
LSTM-5 & 0.68 & 0.93 & 0.017 & 0.9 & 0.958\\
LSTM-1 & 0.63 & 0.11 & 0.77 & 0.04 & \\
LSTM-2 & 0.05 & 0.8 & 0.02 & \\
LSTM-3 & 0.04 & 0.48 & \\
LSTM-4 & 0.01 & \\
LSTM-5 & \\
```

Figura 57. **Estadísticos para los cinco mejores modelos.** Tomado de Python.

Finalmente se escoge al mejor modelo y para determinar su capacidad de generalización se compara con tres enfoques clásicos que se han utilizado comúnmente en la detección de fallos, estos son: Random Forest (RF), árboles de clasificación y regresión (CART) y K-vecinos más cercanos (KNN). Como se indica en la Figura 58.



```
titan@titan-All-Series:~/software/RNN_Fault_Adri/src$
titan@titan-All-Series:~/software/RNN_Fault_Adri/src$ python3 other_Classifie
/usr/local/lib/python3.5/dist-packages/sklearn/ensemble/weight_boosting.py:29
be imported. It will be removed in a future NumPy release.
from numpy.core.umath tests import inner1d
RFmean: 0.5615384615384615 0.045072809823710154
CARTmean: 0.6000000000000002 0.026148818018424558
KNNmean: 0.358974358974359 0.0
KNNmean: 0.41025641025641024 0.0
KNNmean: 0.41025641025641024 0.0
KNNmean: 0.3333333333333337 5.551115123125783e-17
KNNmean: 0.41025641025641024 0.0
KNNmean: 0.38461538461538464 0.0
KNNmean: 0.358974358974359 0.0
KNNmean: 0.3333333333333337 5.551115123125783e-17
KNNmean: 0.28205128205128205 0.0
titan@titan-All-Series:~/software/RNN_Fault_Adri/src$
```

Figura 58. Comparación para con otros enfoques. Tomado de Python

4.3.5 Programación para la evaluación del modelo

Finalmente, para la evaluación del modelo se propone la programación mostrada en la Figura 59.

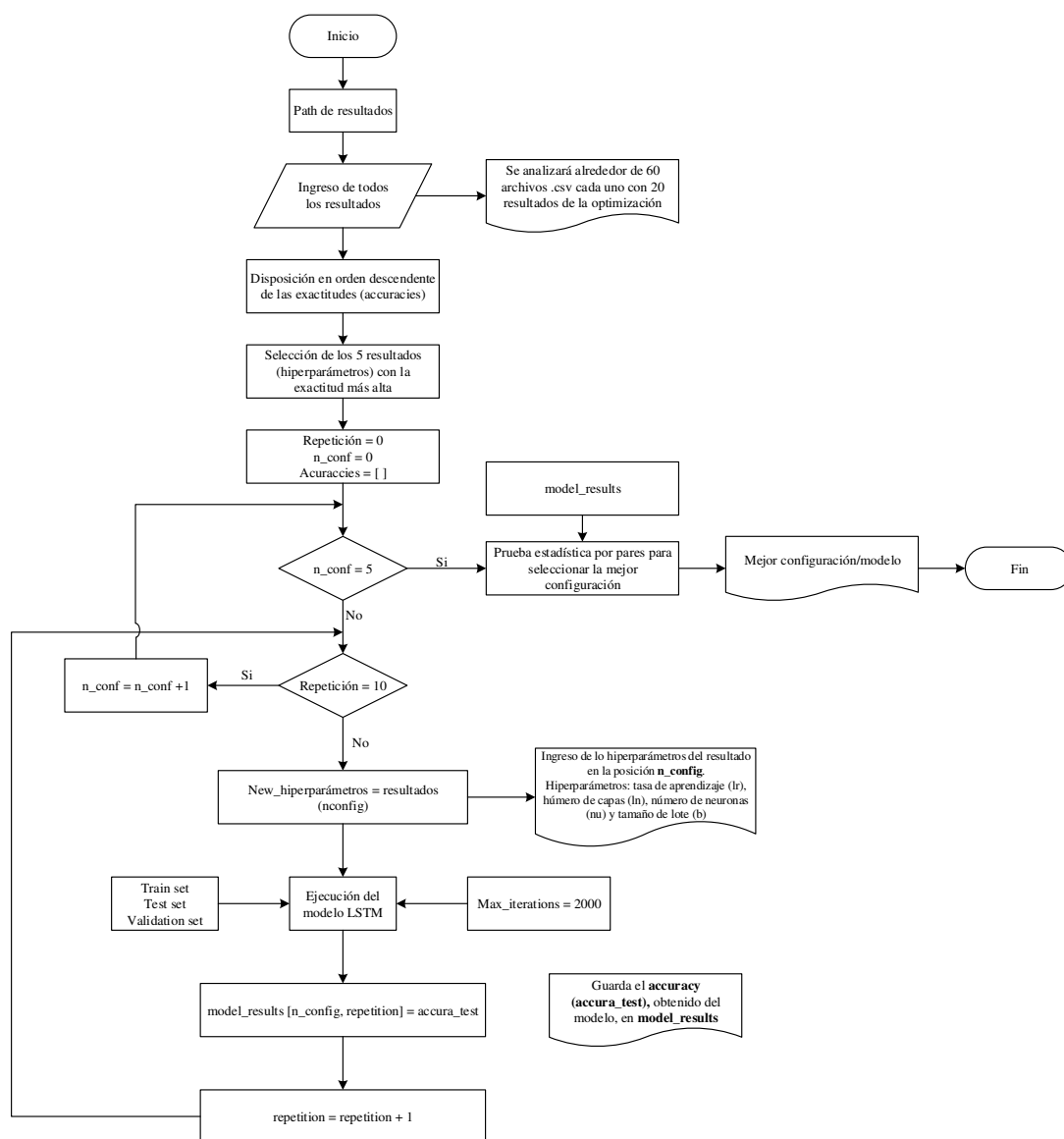


Figura 59. Programación para la evaluación del modelo. Fuente: Propia.

CAPÍTULO 5. RESULTADOS Y DISCUSIÓN

5.1 Análisis e interpretación de resultados de la adquisición de datos y del pre-procesamiento.

Del proceso de adquisición se obtuvo un total de 255 señales en el dominio tiempo. Esta cantidad proviene del número de pruebas efectuadas por cada tipo de fallo (clase) establecidas en la configuración experimental que se propuso en el capítulo 4. De acuerdo con esto, se tienen cinco modos de fallo en las válvulas de ingreso y de descarga, con 17 condiciones de la máquina y 15 repeticiones por cada una. Con este conjunto de señales se aplicó la fase de pre-procesamiento de acuerdo a la metodología de la sección 4.2. Para tener una idea de la incidencia del pre-procesamiento en el incremento del tamaño de los datos a partir de los valores de los parámetros mostrados, se toma una señal que se adquirió en la experimentación con longitud $L = 500\,000$ samples (mediciones) y se establecen valores de $w = 500$, $v = 250$ y $l = 100$ para la aplicación de la ecuación 42, obteniendo como resultado 1898 subseñales. Considerando que en el proceso de adquisición se tomaron 255 señales de vibración, que fueron divididas en los subconjuntos de entrenamiento, validación y prueba con los porcentajes de 70%, 15% y 15% respectivamente, se tiene una cantidad de 179 señales en el conjunto de entrenamiento, 38 en el conjunto de validación y 38 en el conjunto prueba, por lo tanto el cálculo se extiende a todas estas señales. Para el efecto se multiplica el valor del incremento obtenido de una sola señal (1898) por la cantidad de subseñales en cada subconjunto, obteniendo los resultados que se muestran en la Figura 60.

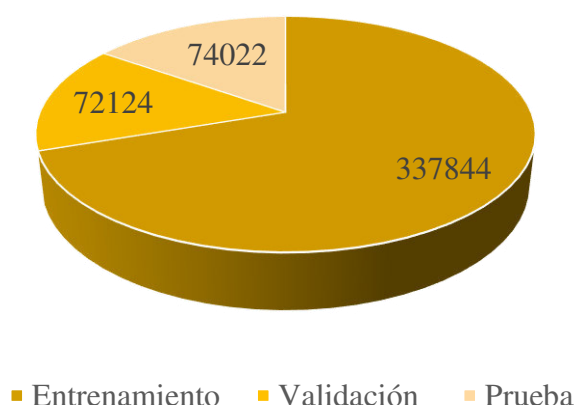


Figura 60. **Cantidad de señales luego del pre-procesamiento.** Elaboración propia.

Como es evidente, el pre-procesamiento tiene gran incidencia en el incremento del tamaño de los datos. Si se suma los valores de los tres subconjuntos se tiene un total de 483 990 señales obtenidas a partir de 255 señales originales. Adicionalmente es preciso indicar que el ejemplo desarrollado es para un solo conjunto de valores de los parámetros de pre-procesamiento. Se diseñaron 60 experimentos en base a los intervalos de prueba de cada parámetro de pre-procesamiento indicados en la Cuadro 18 para verificar su incidencia. Para el efecto se aplicaron 6 valores de longitud de la ventana de atributos (w), 2 valores para la ventana deslizante (v) y 5 valores para la ventana de corte (l). Los resultados de esta experimentación se utilizan para la búsqueda de los mejores hiperparámetros.

Cuadro 18. **Intervalos para los parámetros de pre-procesamiento**

Parámetros	Mínimo valor	Máximo Valor	Incremento
w	500	1000	100
v	$w/2$	w	$w/2$
l	100	180	20

Fuente: Propia

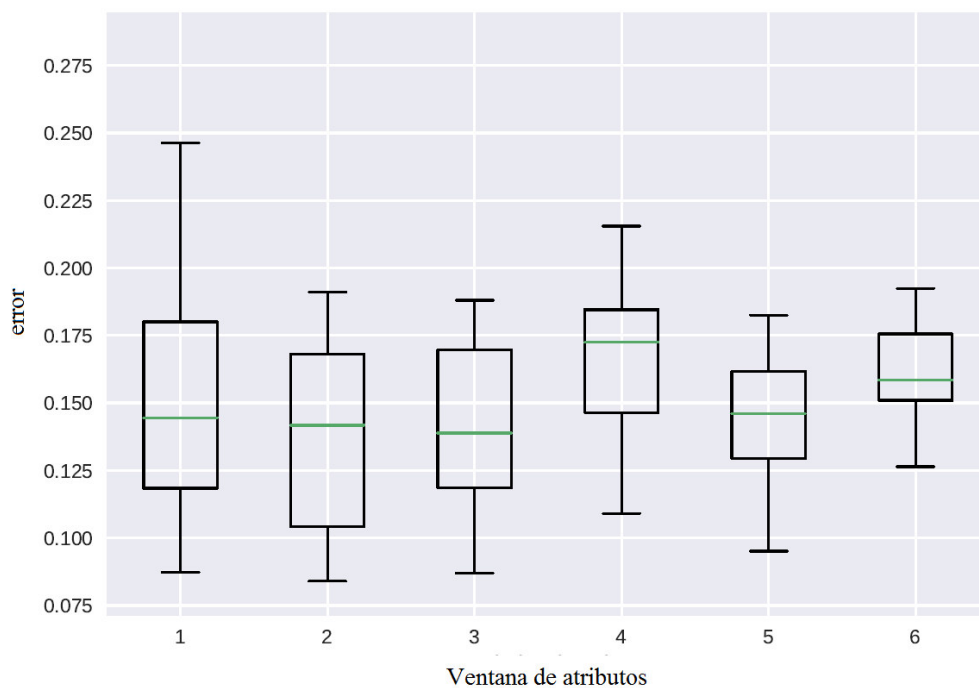
El proceso de búsqueda de los mejores hiperparámetros consta de la evaluación de un modelo LSTM generado a partir de los mismos, para esto se utilizan los datos de validación. El entrenamiento de la red neuronal LSTM con el algoritmo Adam se realizó para esta fase de búsqueda con 200 iteraciones. El número de repeticiones del proceso de optimización de hiperparámetros se fijó en 20, obteniendo como resultado 20 modelos creados en cada experimento.

La Figura 61 (a) y (b) muestran la incidencia de la ventana de atributos (w) y de la ventana de corte (l) en los resultados del modelamiento, respectivamente. Cada diagrama de cajas se calculó con el vector de error resultante obtenido al fijar w (Figura a) y l (Figura b) y variando los otros parámetros. Por ejemplo el vector de errores para la ventana de atributos a 500 mediciones se obtiene variando la ventana deslizante y la ventana de corte en sus diferentes intervalos.

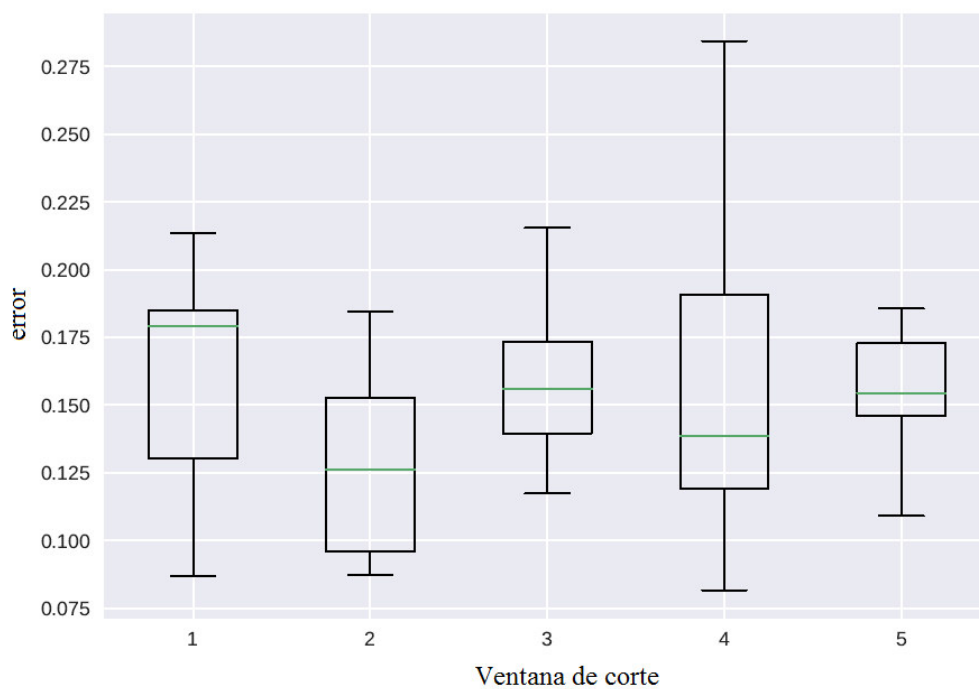
Los resultados muestran menor variabilidad del error con 1000 mediciones de la ventana de atributos que se traduce en su convergencia a un valor medio alrededor de 0.16. Por otro lado el menor error reportado se presenta con una ventana de atributos de 500 y 600 mediciones, con errores más bajos de 0.09 y 0.088, respectivamente. Es evidente que la tendencia de la ventana de corte no es del todo clara, pues el menor error (0.085) se obtiene nuevamente con el valor del parámetro que presenta mayor variabilidad como es el caso de la longitud de la ventana de corte a 160 mediciones.

Para tener mayor contundencia en la definición del mejor modelo, se realizaron otros estudios. Ahora el análisis se centra en los hiperparámetros que corresponden a la mejor arquitectura de la red neuronal. Para el efecto se seleccionaron los cinco mejores hiperparámetros a partir de los resultados anteriores.

En el Cuadro 19 se puede identificar que cuatro de los cinco modelos utilizan valores de 500 y 600 samples (mediciones) de w . Además, se puede notar que los conjuntos de hiperparámetros HP-1, HP-3 Y HP-5 tienen los mismos valores de b , lr , ln y nu ; por lo tanto, tienen la misma configuración, independientemente de los parámetros de pre-procesamiento. Con este resultado se puede decir que un ajuste rápido de 20 iteraciones no es suficiente para determinar los valores de los parámetros de pre-procesamiento que orienten al mejor modelo neuronal.



(a) Venta de atributos de 500, 600, 700, 800, 900 y 1000 mediciones



(b) Ventana de corte de 100, 120, 140, 160 y 180 mediciones

Figura 61. Comparación del error con respecto a los parámetros de pre-procesamiento. (A menor valor favorable). Elaboración propia

Cuadro 19. **Conjunto de modelos seleccionados para comparación de hiperparámetros**

Red	<i>b</i>	<i>lr</i>	<i>nl</i>	<i>nu</i>	<i>w</i>	Δs	<i>l</i>	error
HP-1	8000	0.0925	2	16	800	800	160	0.0815
HP-2	17000	0.0372	3	16	600	300	180	0.0840
HP-3	8000	0.0925	2	16	500	250	120	0.0873
HP-4	14000	0.1011	2	10	500	500	100	0.1879
HP-5	8000	0.0925	2	16	500	250	100	0.2135

Fuente. Propia

5.1.1 Prueba de la hipótesis específica 1

La primera hipótesis específica que se planteó al inicio de la investigación es la siguiente:

- a. **El pre-procesamiento de las señales adquiridas, influye en la construcción de un modelo basado en datos para la detección de fallos de un compresor recíprocante de simple efecto doble etapa.**

Para contrastar esta hipótesis se evalúa el mejor modelo LSTM-5 con dos modelos creados a partir de datos sin efectuar el pre-procesamiento. Se llamará LSTM-S al modelo que se genera a partir de cortes en intervalos de 100 mediciones de su señal, de esta forma la señal no tiene baja resolución pues se toma toda la información contenida en el corte. Por otro lado se tiene al modelo LSTM-R que toma 100 atributos de cada corte de 100 mediciones. Los resultados que se muestran en el Cuadro 20 evidencian una clara ventaja entre el modelo generado a partir de datos pre-procesados, este modelo presenta un mejor rendimiento comparado con los otros dos y también tiene menor variabilidad.

Cuadro 20. Comparación de modelos con y sin pre-procesamiento

Modelo	Exactitud promedio	Desviación estándar
LSTM-S	0,213	0.067
LSTM-R	0,228	0.017
LSTM-5	0.930	0.017

Fuente: Propia.

Los resultados mostrados requieren mayor contundencia para verificar la hipótesis planteada, por lo tanto se aplica la prueba de Anova. Para el efecto, se plantea las dos siguientes hipótesis estadísticas:

Hipótesis nula H_0 : Todos los modelos son estadísticamente iguales.

Hipótesis alternativa H_1 : Todos los modelos son estadísticamente diferentes.

Los resultados que se muestran en Cuadro 21, revelan un valor de significancia de 0.000 que es mucho menor que el nivel de significancia propuesto para la prueba (0.05), por lo tanto se rechaza la hipótesis nula y se acepta la hipótesis alternativa, es decir, todos los modelos son estadísticamente diferentes. Por lo tanto, basándose en los estadísticos de la exactitud promedio y en la prueba Anova, se comprueba que la señal adquirida requiere ser pre-procesada para la construcción de un modelo basado en datos, pues su influencia es evidente.

Cuadro 21. Anova de un factor para los tres grupos.

	Suma de cuadrados	Gl	Media cuadrática	F	Sig.
Inter-grupos	3.362	2	1.681	1005.498	.000
Intra-grupos	.45	27	.002		
Total	3.407	29			

Fuente: Propia

5.2 Análisis e interpretación de resultados de la creación del modelo.

A partir del análisis del conjunto de parámetros de pre-procesamiento, ahora se procede a evaluar cada uno de los cinco mejores conjuntos de hiperparámetros creando modelos LSTM a partir de ellos con una profundidad mayor de entrenamiento, configurada en 2000 iteraciones. Considerando la susceptibilidad de los modelos basados en redes neuronales al momento de establecer sus pesos iniciales, se generaron 10 modelos para cada conjunto HP-i que son evaluados en el subconjunto de datos de prueba. Para el efecto, se aplicó la prueba de Kolmogorov-Smirnov para una hipótesis de distribución gaussiana con un valor de significancia de 0.05. Según el comportamiento gaussiano la caracterización de los resultados se pueden representar por los parámetros: exactitud promedio y desviación estándar. Los resultados se muestran en el Cuadro 22, donde la segunda columna contiene el p-valor para la prueba K-S para cada modelo LSTM-i, configurado a partir del conjunto HP-i de hiperparámetros.

Cuadro 22. Prueba Kolmogorov-Smirnov y métricas estadísticas de precisión para cada modelo LSTM

Modelo	P-valor de la Prueba K-S	Exactitud Promedio	Desviación Estándar
LSTM-1	0.57	0.907	0.025
LSTM-2	0.73	0.901	0.027
LSTM-3	0.96	0.924	0.017
LSTM-4	0.93	0.904	0.021
LSTM-5	0.68	0.930	0.017

Fuente: Propia.

El resultado de la prueba de Kolmogorov-Smirnov muestra que el p-valor para todos los modelos es mayor que 0.05, por lo tanto se acepta la hipótesis que los resultados de todos los modelos siguen una distribución gaussiana. Además de este indicador, se verifica que la exactitud promedio está sobre 0.901 con una variabilidad máxima de 0.027. Para tener una mejor apreciación de estos resultados se ilustra en la Figura 62 los diagramas de cajas que se construyen a partir de los resultados de cada modelo.

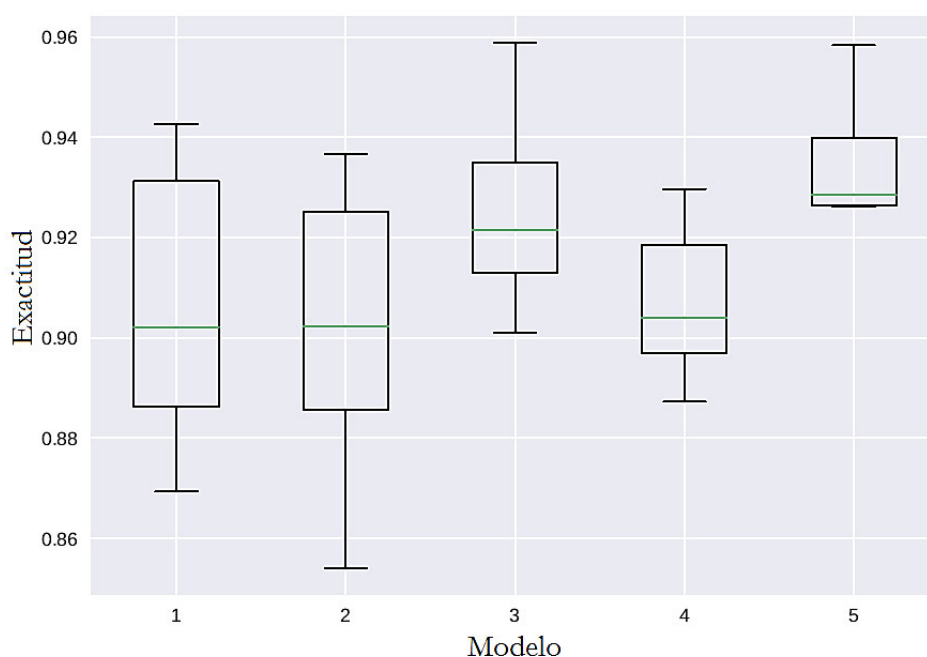


Figura 62. Metodología para la construcción de modelos LSTM a partir de señales de vibración para el diagnóstico de fallas de un compresor (A mayor valor favorable). Elaboración propia.

No obstante, aún no es suficiente evidencia de diferencia estadística para seleccionar al mejor modelo. Para solventar este inconveniente se aplica la prueba T para cada par de modelos, según la hipótesis nula H_0 : todos los pares de modelos son estadísticamente iguales y la hipótesis alternativa H_1 : No todos los pares de modelos son estadísticamente iguales. Los resultados que se muestran en el Cuadro 23 determinan que no todos los pares comparados son iguales, por lo tanto se acepta la hipótesis

alternativa. De este modo se identifican a los pares (LSTM-1, LSTM-2), (LSTM-1, LSTM-3), (LSTM-1, LSTM-4), (LSTM-2, LSTM-4) y (LSTM-3, LSTM-5) como estadísticamente similares pues tienen igual rendimiento. Se puede apreciar que el par (LSTM-3, LSTM-5) presenta el mejor rendimiento con exactitud promedio de 0.924 y 0.930 respectivamente y con igual configuración en su arquitectura al tener los mismos hiperparámetros, la diferencia tan solo radica en la longitud de la subseñal con $l = 120$ para el modelo LSTM-3 y $l = 100$ para LSTM-5. Por otro lado, si se analiza los resultados del error en la fase de optimización de hiperparámetros, el modelo LSTM-3 tiene una clara ventaja con un valor de 0.083 muy por debajo del modelo LSTM-5 que tiene un error 0.2135, concluyendo que la convergencia rápida para 200 iteraciones no es evidencia para seleccionar el mejor modelo.

Cuadro 23. P-Valor de la prueba T entre pares de modelos.

Modelo	LSTM-2	LSTM-3	LSTM-4	LSTM-5
LSTM-1	0.63	0.11	0.77	0.04
LSTM-2	-	0.05	0.8	0.02
LSTM-3	-	-	0.04	0.48
LSTM-4	-	-	-	0.01

Fuente: Propia.

5.2.1 Prueba de la hipótesis específica 2

La segunda hipótesis que se planteó, está enfocada en la optimización de los hiperparámetros y se detalla a continuación:

- b. El proceso de optimización de hiperparámetros mejora la exactitud del modelo diseñado para la detección de fallos del compresor recíprocante de simple efecto doble etapa.**

Para la verificación de esta hipótesis se plantea una comparación entre el mejor modelo LSTM-5 y otro modelo LSTM-SO creado sin la optimización de los hiperparámetros. Para el efecto se aplicó la prueba T-Student para este par de modelos. Para lo cual se plantea las siguientes hipótesis estadísticas:

Hipótesis nula H_0 : Los dos modelos son estadísticamente iguales

Hipótesis alternativa H_1 : Los dos modelos son estadísticamente diferentes

Si se observa en el Cuadro 24, los dos modelos superan el 0.90 de exactitud con una leve ventaja para el modelo LSTM-5, no obstante el resultado de la prueba determina un p-valor de 0.00032358 (<0.05) por lo tanto se rechaza la hipótesis nula y se acepta la hipótesis alternativa que sostiene que los modelos son estadísticamente diferentes. De esta forma se da sustento a la hipótesis de investigación que establece que se requiere un proceso de optimización de hiperparámetros para mejorar la exactitud del modelo diseñado,

Cuadro 24. Prueba T para el par de modelos con y sin optimización de hiperparámetros.

Modelo	Exactitud promedio	P-valor
LSTM-SO	0.903	0.00032358
LSTM-5	0.930	

Fuente: Propia

5.3 Análisis e interpretación de resultados de la generalización del modelo.

Finalmente, para evaluar el modelo y medir su capacidad de generalización, se compara el mejor modelo (LSTM-5) con los clasificadores clásicos Random Forest

(RF), árboles de clasificación y regresión (CART) y K-vecinos más cercanos (KNN). Para una comparación objetiva se utilizaron iguales condiciones de la propuesta, es decir los mismos indicadores de condición como variables de entrada y se crearon 10 modelos para cada enfoque.

Los resultados se muestran en el Cuadro 25 y es claro que el modelo LSTM-5 tiene mayor exactitud en el reconocimiento de patrones temporales frente a los enfoques clásicos, evidenciando la presencia de patrones temporales de indicadores de condición que no han sido capturados por estos modelos; por lo tanto el rendimiento aumenta cuando la característica de alto nivel se extrae de una representación de series temporales mediante el uso del modelo LSTM capaz de capturar los patrones temporales.

Cuadro 25. Comparación entre la propuesta y otros enfoques

Enfoque	Exactitud promedio	Desviación estándar
RF	0.574	0.054
CART	0.590	0.020
KNN	0.410	0.0
LSTM-5	0.930	0.017

Fuente: Propia.

Además del análisis comparativo del mejor modelo LSTM-5 con otros enfoques clásicos es necesario efectuar el análisis de las métricas propias del modelado neuronal. De acuerdo a las secciones 2.3.8 y 4.3.1, para medir la eficiencia del aprendizaje del modelo se necesita la matriz de confusión que se muestran en el Cuadro 26 y las métricas: precisión, recall y f-score del Cuadro 27.

Cuadro 26. Matriz de confusión del modelo LSTM-5 para el análisis de métricas

Fallos	\hat{P}_1	\hat{P}_2	\hat{P}_3	\hat{P}_4	\hat{P}_5	\hat{P}_6	\hat{P}_7	\hat{P}_8	\hat{P}_9	\hat{P}_{10}	\hat{P}_{11}	\hat{P}_{12}	\hat{P}_{13}	\hat{P}_{14}	\hat{P}_{15}	\hat{P}_{16}	\hat{P}_{17}
P_1	161	0	0	0	0	0	0	0	24	0	0	0	5	0	0	0	0
P_2	4	185	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_3	0	0	162	0	23	0	5	0	0	0	0	0	0	0	0	0	0
P_4	0	0	0	178	10	0	0	2	0	0	0	0	0	0	0	0	0
P_5	0	0	2	0	181	0	7	0	0	0	0	0	0	0	0	0	0
P_6	0	0	0	0	0	278	0	0	7	0	0	0	0	0	0	0	0
P_7	0	0	13	1	35	0	141	0	0	0	0	0	0	0	0	0	0
P_8	0	0	0	0	0	0	19	169	0	0	0	0	2	0	0	0	0
P_9	6	0	0	0	0	38	0	0	241	0	0	0	0	0	0	0	0
P_{10}	1	0	0	0	0	0	0	2	0	276	0	0	5	0	1	0	0
P_{11}	0	0	0	0	0	0	0	0	0	0	190	0	0	0	0	0	0
P_{12}	0	0	0	0	0	0	0	0	0	0	2	188	0	0	0	0	0
P_{13}	0	0	0	0	0	0	0	0	0	0	0	0	284	0	0	1	0
P_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	190	0	0	0
P_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	187	3	0
P_{16}	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	269	0
P_{17}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	190

Fuente: Propia.

Cuadro 27. Métricas para medir la eficiencia del modelo

Condición de la máquina	Precisión	Recall	F-score
P1	0.94	0.85	0.89
P2	1.00	0.97	0.99
P3	0.92	0.85	0.88
P4	0.99	0.94	0.96
P5	0.73	0.95	0.82
P6	0.88	0.98	0.93
P7	0.82	0.74	0.78
P8	0.98	0.89	0.93
P9	0.89	0.85	0.87
P10	1.00	0.97	0.98
P11	0.99	1.00	0.99
P12	1.00	0.99	0.99
P13	0.91	1.00	0.95
P14	0.99	1.00	1.00
P15	0.99	0.98	0.99
P16	0.99	0.94	0.96
P17	1.00	1.00	1.00

Fuente: Propia.

Recordando, la precisión es la capacidad del modelo para evitar clasificar incorrectamente muestras que no sean del k-ésimo modo de fallo como k-ésimo modo de fallo, en tanto que el recall es la capacidad del modelo para evitar clasificar incorrectamente muestras del k-ésimo modo de fallo como no k-ésimo modo de fallo y el f-score es la relación entre la precisión y el recall. En los cuadros anteriores se presentan la matriz de confusión y el cálculo de las métricas para el modelo LSTM-5 respectivamente. En la matriz de confusión constan 17 condiciones de la máquina de acuerdo a la experimentación que se estableció. A continuación se analizan las métricas para cada condición de la máquina, considerando que el análisis para la precisión se efectúa en sentido vertical y para el recall el análisis es horizontal.

- Condición P1 (Condición normal de todos los componentes): Para esta condición se tiene que cuatro clases de la condición P2 han sido clasificadas como condición P1, esto da un valor de precisión de 0.94. No así para el recall, donde se tienen 24 clases que perteneciendo a la condición P1 se clasificaron como P9 dando como resultado un recall de 0.85. El análisis determina que para la condición de análisis P1, la precisión es mayor que el recall, debido a la clasificación descrita.
- Condición P2 (Desgaste del asiento de la válvula de descarga en la segunda etapa.): Para esta condición todas los modos de fallo han sido clasificados correctamente a su correspondiente, por lo tanto se tiene una precisión de 1. Para el recall se tiene un valor de 0.97 debido a que el modelo asigna erróneamente 4 modos de fallo de a la condición P1 y un modo de fallo a la condición P14.
- Condición P3 (Corrosión de la válvula de descarga en la segunda etapa): En esta condición se tiene una precisión de 0.92, pues la cantidad de modos de fallos asignados erróneamente a la condición P3 son 2 del P5 y 13 del P7. Para el recall se tiene un valor de 0.85 debido a que se asignan 23 modos de fallo de la condición P3 a la condición P5 y 5 a la condición P7.
- Condición P4 (Fractura del plato de la válvula de descarga en segunda etapa): Para esta condición tanto la precisión como el recall superan el 0.90. Este resultado favorable se debe a que únicamente 1 modo de fallo de la condición P7 ha sido

asignado incorrectamente a la condición P4. Y para el recall se tiene que 12 modos de la condición P4 se han asignado a otras condiciones.

- Condición P5 (Rotura de resorte de la válvula de descarga en la segunda etapa.): para esta condición se tiene el valor más bajo de la clasificación 0.73, si analizamos detectamos que han sido asignados erróneamente a esta condición 35 modos de fallos que pertenecen a la condición P7. Por otro lado el recall supera el 0.90 pues 9 modos de fallo de esta clase han sido clasificados erróneamente a otras condiciones de la máquina.
- Condición P6 (Desgaste del asiento de la válvula de succió en la segunda etapa.): Para esta condición también se tiene un valor de la precisión por debajo de 0.90, el motivo lo originan 38 modos de fallo de P9 asignados a la condición P6, sin embargo para el cálculo del recall se tiene que únicamente 7 modos de fallo de la P6 se asignaron erróneamente a P9, dando un valor de 0.98.
- Condición P7 (Corrosión del plato de la válvula de admisión en la segunda etapa.): En esta condición también se observan valores bajos tanto de la precisión (0.82) como del recall (0.74). Para el precisión se nota que 31 modos de fallo de otras clases fueron asignadas erróneamente a la clase P7 y que 49 modos de fallos que pertenecen a las condición P7 se asignaron a otras clases.
- Condición P8 (Fractura del plato dela válvula de succión en la segunda etapa.): En esta condición la precisión es alta (0.98), pues tan sólo cuatro modos de fallo fueron asignados erróneamente a esta condición. El recall tiene un valor de 0.89 porque 21 modos de fallo de esta condición se asignaron a otras condiciones.
- Condición P9 (Rotura de resorte de la válvula de succión en la segunda etapa.): La precisión tiene un valor de 0.89 porque 31 modos de fallos de otras condiciones fueron asignados a la condición P9 y el recall se valora en 0.85 porque 44 modos de fallo de la clase P9 se asignaron a otras clases.
- Condición P10 (Desgaste del asiento de la válvula de descarga en la primera etapa.): En esta condición la clasificación para el cálculo de la precisión ha sido

correcta porque se tiene una precisión de 1; el recall también tiene un valor alto (0.97) pues cuatro modos de fallo de la condición P10 se ha asignado a otras condiciones.

- Condición P11 (Corrosión del plato de la válvula de descarga en la primera etapa.): En esta condición tanto la precisión (0.99) como el recall (1) tienen valores altos. Se encuentra que únicamente dos modos de fallo de la clase P12 se asignaron erróneamente a esta clase.
- Condición P12 (Fractura del plato de la válvula de descarga en primera etapa.): al igual que en la anterior la precisión (1) y el recall (0.99) tiene valores altos. Se observa que dos muestras de la condición P12 fueron asignadas a la condición P11.
- Condición P13 (Rotura de muelle de la válvula de descarga en primera etapa.): La precisión para esta condición es de 0.91 porque 5 modos de fallo de la condición P1, 2 de la P8 y 5 de la P10 se asignaron erróneamente a la clase P13. El recall tiene un valor de 1.
- Condición P14 (Desgaste del asiento de la válvula de succión N°2 en la primera etapa.): Para esta condición la precisión tiene un valor de 0.99 pues se evidencia que un modo de fallo de P2 se asignó a P14. Por otra parte el recall tiene un valor de 1, demostrando que la clasificación para este valor fue exacta.
- Condición P15 (Corrosión del plato de la válvula de succión N°2 en la primera etapa.): Se evidencia que en esta condición se ha asignado incorrectamente un modo de fallo de otra condición, por lo tanto la precisión es de (0.99) y el recall se valora en 0.98 porque se evidencia que tres modos de fallo se de P15 se clasificaron como P16.
- Condición P16 (Fractura del plato de la válvula de succión N°2 en la primera etapa.): En esta condición la precisión es de 0.99 porque 1 modo de fallo de P13 y 3 de P15 fueron asignados a P16. El recall tiene un valor de 0.94 porque 16 muestras se asignaron erróneamente a P13.

- Condición P17 (Rotura de resorte de válvula de succión N°2 en la primera etapa.): Para esta condición tanto el valor de la precisión como del recall tiene un valor de 1, es decir la clasificación fue correcta.

Si analizamos los resultados podemos notar que los valores más bajos de la precisión se originan en las condiciones P5, P6, P7 y P9 pues están por debajo de 0.9. Para el recall los valores más bajos se tienen en las condiciones P1, P3, P5, P7 y P9. Se puede evidenciar que los valores bajos se repiten en las condiciones P5, P7 y P9.

Si se analiza el F-score se puede evidenciar que las condiciones con menor valor son la P1, P3, P5, P7 y P9, por lo tanto indica que el modelo no se desenvolvió de manera eficiente para estas condiciones, pues si tomamos en consideración que determina un promedio o relación entre la precisión y el recall. Este valor es una medida de la exactitud del modelo al momento de clasificar. Los resultados que se presentaron y analizaron en esta sección, sirven de base para la contrastación de las hipótesis que se plantearon.

5.3.1 Prueba de la hipótesis específica 3

La tercera y última hipótesis específica planteada, se enfoca en la generalización del modelo que a continuación se detalla y analiza.

- c. **La evaluación del modelo generado a partir de señales de vibración es capaz de generalizar la detección de fallos en el compresor reciprocante de simple efecto doble etapa.**

Para la verificación de esta hipótesis, una vez escogidos los cinco mejores modelos se realizaron diferentes pruebas para comprobar el mejor rendimiento. Producto de ello se estableció que los cinco modelos superan el p-valor que orienta a determinar

que todos los modelos LSTM-i son aptos para la detección de fallos. Sin embargo para tener mayor contundencia en la aceptación de esta hipótesis, nos centramos en la comparación del mejor modelo LSTM-5 con tres enfoques tradicionales. La mejor exactitud promedio fue de 0.93 para el modelo neuronal considerado valor óptimo en comparación con tres enfoques tradicionales que se muestran en el Cuadro 25. Adicionalmente se realiza la prueba Anova para comparar los cuatro grupos de modelos RF, CART, KNN y LSTM para sustentar aún más la hipótesis de investigación. Para lo cual se plantean las siguientes hipótesis estadísticas:

Hipótesis nula H_0 : Los dos modelos son estadísticamente iguales

Hipótesis alternativa H_1 : Los dos modelos son estadísticamente diferentes

Los resultados de esta prueba demuestran un valor de significancia de $0.000 < 0.05$, dando lugar al rechazo de la hipótesis nula y aceptación de la hipótesis alternativa, que afirma que los modelos analizados son estadísticamente diferente. Este resultado orienta a su vez a la comprobación de la hipótesis específica tres que enuncia que el modelo generado tiene capacidad de generalización, es decir puede ser aplicable a la detección de fallos en el compresor recíprocante de simple efecto doble etapa. Los resultados se resumen en el Cuadro 28.

Cuadro 28. **Prueba Anova para comprobación de hipótesis tres**

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	1.607	3	0.536	673.407	0.000
Intra-grupos	0.029	36	0.001		
Total	1.636	39			

Fuente: Propia.

CONCLUSIONES

Con la realización del presente proyecto se determinó que es posible crear un modelo basado en datos de vibración para la detección de fallos en compresores reciprocantes. El enfoque de redes neuronales artificiales recurrentes del tipo LSTM demostró ser altamente eficiente con respecto a otros enfoques tradicionales utilizados hasta el momento. La diferencia sustancial en la exactitud de reconocimiento de patrones temporales del modelo LSTM lleva a concluir que hasta el momento es la mejor opción para sistemas dinámicos donde se requiere un reconocimiento de patrones temporales a corto y largo plazo.

Conclusión específica 1

Se comprobó que el desarrollo de un pre-procesamiento de las señales adquiridas influye en la construcción de un modelo basado en datos para la detección de fallos de un compresor reciprocante de simple efecto doble etapa, pues se evidenció que la exactitud promedio del modelo con optimización es superior al modelo generado sin optimización. Esto fue posible con el diseño de una guía para la adquisición de señales que permitió determinar cuáles son los aspectos claves que intervienen para una correcta adquisición, entre las que se tiene: la configuración del compresor, el tipo de sensor utilizado, su ubicación, el software utilizado y el medio donde se ubica la máquina. Por otro lado se planteó una metodología para el pre-procesamiento de la señal, destacando principalmente su gran utilidad para el tratamiento de las series de tiempo y siendo aplicable no sólo a compresores sino a cualquier tipo de maquinaria.

Conclusión específica 2

Se comprobó que es necesario realizar un proceso de optimización de hiperparámetros que son propios de la arquitectura neuronal para alcanzar un diseño adecuado que permita obtener la mayor exactitud del modelo. Al momento de realizar

el mapeo de las características para la detección de fallos en el compresor recíprocante se pudo comprobar que el modelo con la aplicación del proceso de la optimización obtuvo más del doble de exactitud del modelo sin optimización. Además se evidenció que la utilización de la optimización bayesiana ahorra el recurso tiempo y carga computacional, pues el proceso iterativo para alcanzar los mejores modelos se realizó de forma prácticamente automática.

Conclusión específica 3

Finalmente se pudo comprobar que el modelo generado tiene alta precisión al momento de realizar el mapeo de las características de fallo, por lo tanto es posible generalizar su aplicación para la detección de fallos en el compresor y en otras máquinas donde sus sistemas sean dinámicos. De esta forma, se puede concluir que el uso del modelo LSTM junto con una representación de baja resolución de una señal de vibración es adecuado para capturar los patrones temporales que permiten realizar la tarea de diagnóstico, ya que evidenció una exactitud superior al 90% para 17 condiciones de falla en comparación con otros modelos tradicionales aplicados al diagnóstico de fallas.

Conclusiones adicionales

La configuración experimental propuesta considera un solo tipo de fallo en cada válvula de ingreso y de descarga, es decir no se tiene un sistema multifallos. Los fallos que se proponen en la investigación han sido efectuados artificialmente para simular las condiciones reales de desgaste.

La Optimización Bayesiana fue un aspecto clave para buscar de manera óptima el mejor conjunto de hiperparámetros que conducen al máximo valor de exactitud en la detección de fallos a través de la información acumulada de cada iteración. El resultado demuestra que luego de 200 iteraciones el comportamiento de las cinco mejores modelos se orienta a iguales conjuntos de hiperparámetros, pues los modelos LSTM-1, LSTM-3 y LSTM-5 tienen iguales valores para b , lr , ln y nu , demostrando así la preferencia por estos valores al momento de alcanzar la convergencia.

Los valores más bajos de precisión y recall se obtuvieron en la condición de rotura del resorte, que conlleva a analizar qué pasa con este fallo para la estructuración del modelo.

Los productos obtenidos de esta investigación son: una tesis de pre-grado titulada “Adquisición de señales acústicas y de vibración para el diagnóstico de fallos en un compresor recíprocante de doble etapa”, una ponencia en el Congreso Latinoamericano de Ingeniería Mecánica (COLIM 2018) realizado en Colombia con el título “Incremento del tamaño de los datos para la detección de fallos en maquinaria rotativa”, donde se puso en práctica la metodología planteada para el pre-procesamiento de los datos. Además la ponencia fue aceptada para su publicación en la revista Bitsua-Colombia. Finalmente se ha presentado un segundo artículo para su publicación, donde se plasmó toda la investigación y con título “Bayesian approach to Long Short-Term Memory based model building for fault diagnosis of a reciprocating compressor”.

RECOMENDACIONES

Las recomendaciones se enfocan en trabajos futuros que se pueden llevar a cabo a partir de la presente investigación y son las siguientes:

- Realizar el mismo procedimiento de la investigación, pero incluyendo multifallos es decir, implementar más de un fallo en el sistema de funcionamiento del compresor y/o combinar fallos.
- Ampliar el análisis de fallos a otros componentes del compresor, como es el caso de los rodamientos.
- Considerar la optimización de los hiperparámetros para una optimización directa junto con los parámetros de pre-procesamiento, pues sería un ahorro computacional al momento de realizar los experimentos.
- Analizar cuáles son las causas de la confusión al momento de la clasificación en la condiciones de la máquina que dieron como resultado valores bajos de la precisión y recall. De esta forma se tendría mayor criterio desde el punto de vista físico para mejorar las condiciones experimentales al momento de la adquisición de las señales de vibración.
- Incursionar en otro tipo de parámetros de monitoreo de la condición de la máquina, como la emisión acústica o ruido.

REFERENCIAS BIBLIOGRÁFICAS

- Ahmad, R., & Kamaruddin, S. (2012a). An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, 63(1), 135-149. <https://doi.org/10.1016/j.cie.2012.02.002>
- Ahmed, M., Baqqar, M., Gu, F., & Ball, A. D. (2012b). Fault detection and diagnosis using Principal Component Analysis of vibration data from a reciprocating compressor. En *Proceedings of 2012 UKACC International Conference on Control* (pp. 461-466). <https://doi.org/10.1109/CONTROL.2012.6334674>
- Ahmed, M., Smith, A., Gu, F., & Ball, A. D. (2014). Fault diagnosis of reciprocating compressors using relevance vector machines with a genetic algorithm based on vibration data. En *2014 20th International Conference on Automation and Computing* (pp. 164-169). <https://doi.org/10.1109/IConAC.2014.6935480>
- Ahmed, Mahmud, Abdusslam, S. A., Baqqar, M., Gu, F., & Ball, A. (2011). Fault Classification of Reciprocating Compressor Based on Neural Networks and Support Vector Machines. En *Proceedings of the 17th International Conference on Automation & Computing*. Huddersfield: Chinese Automation and Computing Society. Recuperado de <http://eprints.hud.ac.uk/11491/>
- Ahmed, R., Sayed, M. E., Gadsden, S. A., Tjong, J., & Habibi, S. (2015). Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques. *IEEE Transactions on Vehicular Technology*, 64(1), 21-33. <https://doi.org/10.1109/TVT.2014.2317736>
- Alaswad, Suzan, & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety*, 157, 54-63. <https://doi.org/10.1016/j.ress.2016.08.009>
- Aravinth, S., Kanna, K. R., & Sugumaran, V. (2016). Air Compressor Fault Diagnosis Through Vibration Signals using Statistical Features and J48 Algorithms. *Indian Journal of Science and Technology*, 9(47).

- Baillie, D. C., & Mathew, J. (1997). Nonlinear model-based fault diagnosis of bearings. *NDT and E International*, 5(30), 328.
- Bayar, N., Darmoul, S., Hajri-Gabouj, S., & Pierreval, H. (2015). Fault detection, diagnosis and recovery using Artificial Immune Systems: A review. *Engineering Applications of Artificial Intelligence*, 46, 43-57. <https://doi.org/10.1016/j.engappai.2015.08.006>
- Belman-Flores, J. M., Ledesma, S., Barroso-Maldonado, J. M., & Navarro-Esbrí, J. (2015). A comparison between the modeling of a reciprocating compressor using artificial neural network and physical model. *International Journal of Refrigeration*, 59(Supplement C), 144-156. <https://doi.org/10.1016/j.ijrefrig.2015.07.017>
- Benavides, G. (2013). *La Ecuación de la vida: Los Cuatro principios basicos para alcanzar una vida plena*. BookBaby.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157-166. <https://doi.org/10.1109/72.279181>
- Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. En *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28* (pp. I-115–I-123). Atlanta, GA, USA: JMLR.org. Recuperado de <http://dl.acm.org/citation.cfm?id=3042817.3042832>
- Bevilacqua, M., & Braglia, M. (2000). The analytic hierarchy process applied to maintenance strategy selection. *Reliability Engineering & System Safety*, 70(1), 71-83. [https://doi.org/10.1016/S0951-8320\(00\)00047-8](https://doi.org/10.1016/S0951-8320(00)00047-8)
- Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). Recurrent Neural Network Architectures. En F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, & R. Jenssen (Eds.), *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis* (pp. 23-

- 29). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-70338-1_3
- Bloch, H. P., & Hoefner, J. J. (1996). *Reciprocating Compressors: Operation and Maintenance*. Elsevier.
- Bunge, M. (2014). *La ciencia, su método y su filosofía*. Penguin Random House Grupo Editorial Argentina.
- Buscema, M. (2013). Supervised Artificial Neural Networks: Backpropagation Neural Networks. En M. Buscema & W. J. Tastle (Eds.), *Intelligent Data Mining in Law Enforcement Analytics: New Neural Networks Applied to Real Problems* (pp. 119-135). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-4914-6_7
- Cajas Muñoz, F. D., & Torres Díaz, C. P. (2018). Adquisición de señales acústicas y de vibración para el diagnóstico de fallos en un compresor reciprocante de doble etapa. Recuperado de <http://dspace.ups.edu.ec/handle/123456789/16592>
- Carbonell, J. G., Michalski, R. S., Mitchell (auth.), T. M., Michalski, R. S., Carbonell, J. G., & Mitchell (eds.), T. M. (1983). *Machine Learning: An Artificial Intelligence Approach* (1.^a ed.). Springer-Verlag Berlin Heidelberg. Recuperado de <http://gen.lib.rus.ec/book/index.php?md5=557f19ba12ff5d33c88aa17871bfb10d>
- Cardenal, A. (2018, febrero 21). BREVE HISTORIA DEL MANTENIMIENTO INDUSTRIAL. Recuperado 20 de agosto de 2018, de <https://alfonsocardenal.wordpress.com/2018/02/21/historia-del-mantenimiento/>
- Chang, C.-C. (2014). Optimum preventive maintenance policies for systems subject to random working times, replacement, and minimal repair. *Computers & Industrial Engineering*, 67, 185-194. <https://doi.org/10.1016/j.cie.2013.11.011>
- Chen, Z., & Lian, X. (2010). Fault diagnosis for valves of compressors based on Support Vector Machine. En *2010 Chinese Control and Decision Conference* (pp. 1235-1238). <https://doi.org/10.1109/CCDC.2010.5498165>

- Choi, G. H., & Choi, G. S. (1996). APPLICATION OF MINIMUM CROSS ENTROPY TO MODEL-BASED MONITORING IN DIAMOND TURNING. *Mechanical Systems and Signal Processing*, 10(5), 615-631. <https://doi.org/10.1006/mssp.1996.0042>
- Clasificación Supervisada y No Supervisada - Fernando Sancho Caparrini. (s. f.). Recuperado 8 de octubre de 2018, de <http://www.cs.us.es/~fsancho/?e=77>
- Coria, V. H., Maximov, S., Rivas-Dávalos, F., Melchor, C. L., & Guardado, J. L. (2015). Analytical method for optimization of maintenance policy based on available system failure data. *Reliability Engineering & System Safety*, 135, 55-63. <https://doi.org/10.1016/j.ress.2014.11.003>
- Cruz, P. P. (2011). *Inteligencia artificial con aplicaciones a la ingeniería*. Marcombo.
- Destrero, A., Mosci, S., Mol, C. D., Verri, A., & Odone, F. (2009). Feature selection for high-dimensional data. *Computational Management Science*, 6(1), 25-40. <https://doi.org/10.1007/s10287-008-0070-7>
- Devendiran, S., & Manivannan, K. (2016). Vibration Based Condition Monitoring and Fault Diagnosis Technologies For Bearing and Gear Components-A Review. *International Journal of Applied Engineering Research*, 11(6), 3966–3975.
- Diferentes tipos de compresores. (s. f.). Recuperado 7 de marzo de 2018, de <https://www.mundocompresor.com/articulos-tecnicos/diferentes-tipos-compresores>
- Elhaj, M., Gu, F., Ball, A. D., Albarbar, A., Al-Qattan, M., & Naid, A. (2008). Numerical simulation and experimental study of a two-stage reciprocating compressor for condition monitoring. *Mechanical Systems and Signal Processing*, 22(2), 374-389. <https://doi.org/10.1016/j.ymssp.2007.08.003>
- Ellman, A., & Piché, R. (1999). Recurrent Neural Networks for Fault Diagnosis and Severity Assessment of a Screw Compressor. *Journal of dynamic systems, measurement, and control*, 121(4), 721–724.

- El-Thalji, I., & Jantunen, E. (2015). A summary of fault modelling and predictive health monitoring of rolling element bearings. *Mechanical Systems and Signal Processing*, 60-61, 252-272. <https://doi.org/10.1016/j.ymssp.2015.02.008>
- Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendiente - Fernando Sancho Caparrini. (s. f.). Recuperado 17 de octubre de 2018, de <http://www.cs.us.es/~fsancho/?e=165>
- Farzaneh-Gord, M., & Khoshnazar, H. (2016). Valve fault detection for single-stage reciprocating compressors. *Journal of Natural Gas Science and Engineering*, 35(Part A), 1239-1248. <https://doi.org/10.1016/j.jngse.2016.09.062>
- Fernandes de Mello, R., & Antonelli Ponti, M. (2018). A Brief Review on Machine Learning. En R. Fernandes de Mello & M. Antonelli Ponti (Eds.), *Machine Learning: A Practical Approach on the Statistical Learning Theory* (pp. 1-74). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-94989-5_1
- Fernandez P. (2000). Compresores. Universidad de Cantabria. Recuperado de <http://es.pfernandezdiez.es/?pageID=6>
- Fugate, M. L., Sohn, H., & Farrar, C. R. (2001). VIBRATION-BASED DAMAGE DETECTION USING STATISTICAL PROCESS CONTROL. *Mechanical Systems and Signal Processing*, 15(4), 707-721. <https://doi.org/10.1006/mssp.2000.1323>
- Galar, D., Thaduri, A., Catelani, M., & Ciani, L. (2015). Context awareness for maintenance decision making: A diagnosis and prognosis approach. *Measurement*, 67, 137-150.
- Goyal, D., & Pabla, B. S. (2016). The Vibration Monitoring Methods and Signal Processing Techniques for Structural Health Monitoring: A Review. *Archives of Computational Methods in Engineering*, 23(4), 585-594. <https://doi.org/10.1007/s11831-015-9145-0>
- Goyal, Deepam, & Pabla, B. S. (2015). Condition based maintenance of machine tools—A review. *CIRP Journal of Manufacturing Science and Technology*, 10, 24-35. <https://doi.org/10.1016/j.cirpj.2015.05.004>

- Goyal, Deepam, Pabla, B. S., Dharni, S. S., & Lachhwani, K. (2017). Optimization of condition-based maintenance using soft computing. *Neural Computing and Applications*, 28(1), 829-844. <https://doi.org/10.1007/s00521-016-2377-6>
- Grajales, D. H. M., Sánchez, Y. O., & Pinzón, M. (2006). La confiabilidad, la disponibilidad y la mantenibilidad, disciplinas modernas aplicadas al mantenimiento. *Scientia et technica*, 1(30). Recuperado de <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/6513>
- Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. En *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6645-6649). <https://doi.org/10.1109/ICASSP.2013.6638947>
- Guerra, C. (2013). *Condition Monitoring of Reciprocating Compressor Valves Using Analytical and Data-Driven Methodologies*. United States: UMT.
- Guerra, C. J., & Kolodziej, J. R. (2014). A Data-Driven Approach for Condition Monitoring of Reciprocating Compressor Valves. *Journal of Engineering for Gas Turbines and Power*, 136(4), 041601. <https://doi.org/10.1115/1.4025944>
- Guiracocha, R. (2015). *Establecimiento de una base de datos de señales de vibraciones acústicas e imágenes termografías infrarrojas para un sistema mecánico rotativo con la combinación de diferentes tipos de fallos y elaboración de guías de prácticas para detección de fallos en engranajes* (Tesis de pregrado). Universidad Politécnica Salesiana, Cuenca, Ecuador.
- Hagan, M. T., Demuth, H. B., & Beale, M. H. (2014). *Neural network design*. Martin Hagan.
- Haiyang, Z., Jindong, W., Hui, H., & Yiqi, G. (2016). A feature extraction method based on HLMD and MFE for bearing clearance fault of reciprocating compressor. *Measurement*, 89, 34-43. <https://doi.org/10.1016/j.measurement.2016.03.076>
- Henao, D. (2000). *Inteligencia artificial*. El Cid Editor | apuntes.

- Hernández, R., Fernández, C., & Baptista, M. (2014). *Metodología de la Investigación* (sexta). McGraw-Hill Interamericana.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Howard, I., Jia, S., & Wang, J. (2001). THE DYNAMIC MODELLING OF A SPUR GEAR IN MESH INCLUDING FRICTION AND A CRACK. *Mechanical Systems and Signal Processing*, 15(5), 831-853. <https://doi.org/10.1006/mssp.2001.1414>
- Hui, K. H., Hee, L. M., Leong, M. S., & Abdelrhman, A. M. (2015). Vibration Condition Monitoring: Latest Trend and Review. *Applied Mechanics and Materials; Zurich*, 773-774, 139-143. <http://bibliotecavirtual.ups.edu.ec:2099/10.4028/www.scientific.net/AMM.773-774.139>
- Instituto de Mecánica de los Fluidos e Ingeniería Ambiental. (2010). Compresores. Universidad de la Republica: Uruguay. Recuperado de https://www.fing.edu.uy/imfia/cursos/maq_flu_1/teorico/8-Compresores.2010.pdf
- Introducción al Aprendizaje Automático - Fernando Sancho Caparrini. (s. f.). Recuperado 3 de octubre de 2018, de <http://www.cs.us.es/~fsancho/?e=75>
- Isermann, R. (2006). *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Science & Business Media.
- ISO 15243:2004. Rolling bearings — Damage and failures — Terms, characteristics and causes.
- James Li, C., & Yu, X. (1995). High pressure air compressor valve fault diagnosis using feedforward neural networks. *Mechanical Systems and Signal Processing*, 9(5), 527-536. <https://doi.org/10.1006/mssp.1995.0040>
- Jardine, Andrew K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance.

- Mechanical Systems and Signal Processing*, 20(7), 1483-1510.
<https://doi.org/10.1016/j.ymssp.2005.09.012>
- Jiang, Z., Zhang, J., Jin, M., & Ma, B. (2013). An expert system based on multi-source signal integration for reciprocating compressor. *WSEAS Transactions on Systems*, 12(5), 266–279.
- Keerqinhu, Qi, G., Tsai, W. T., Hong, Y., Wang, W., Hou, G., & Zhu, Z. (2016). Fault-Diagnosis for Reciprocating Compressors Using Big Data. En *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 72-81).
<https://doi.org/10.1109/BigDataService.2016.27>
- Kostyukov, V. N., & Naumenko, A. P. (2016). About the Experience in Operation of Reciprocating Compressors Under Control of the Vibration Monitoring System. *Procedia Engineering*, 152(Supplement C), 497-504.
<https://doi.org/10.1016/j.proeng.2016.07.635>
- Kothamasu, R., Huang, S. H., & VerDuin, W. H. (2006). System health monitoring and prognostics — a review of current paradigms and practices. *The International Journal of Advanced Manufacturing Technology*, 28(9-10), 1012-1024. <https://doi.org/10.1007/s00170-004-2131-6>
- Kothamasu, R., Shi, J., Huang, S. H., & Leep, H. R. (2004). Comparison of Selected Model Evaluation Criteria for Maintenance Applications. *Structural Health Monitoring*, 3(3), 213-224. <https://doi.org/10.1177/1475921704042696>
- Kuhn, S. T. (2011). *La estructura de las revoluciones científicas*. Fondo de Cultura Económica.
- Kumar, A., Shankar, R., & Thakur, L. S. (2017). A big data driven sustainable manufacturing framework for condition-based maintenance prediction. *Journal of Computational Science*. <https://doi.org/10.1016/j.jocs.2017.06.006>
- Lahoz-Beltrá, R. (2004). *Bioinformática: simulación, vida artificial e inteligencia artificial*. Ediciones Díaz de Santos.

- Landa J. (2014). *Pistones para motores térmicos y compresores alternativos: diseño, cálculo, simulación y ensayos*. Pais Vasco.
- León, F. C. G. de. (1998). *Tecnología del mantenimiento industrial*. EDITUM.
- Li, C., Sanchez, R.-V., Zurita, G., Cerrada, M., Cabrera, D., & Vásquez, R. E. (2016a). Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical Systems and Signal Processing*, 76-77, 283-293. <https://doi.org/10.1016/j.ymssp.2016.02.007>
- Li, C., Sanchez, V., Zurita, G., Cerrada Lozada, M., & Cabrera, D. (2016b). Rolling element bearing defect detection using the generalized synchrosqueezing transform guided by time–frequency ridge enhancement. *ISA Transactions*, 60, 274-284. <https://doi.org/10.1016/j.isatra.2015.10.014>
- Li, C., Valente de Oliveira, J., Cerrada, M., Pacheco, F., Cabrera, D., Sanchez, V., & Zurita, G. (2016c). Observer-biased bearing condition monitoring: From fault detection to multi-fault classification. *Engineering Applications of Artificial Intelligence*, 50, 287-301. <https://doi.org/10.1016/j.engappai.2016.01.038>
- Liang, Z., Li, S., Tian, J., Zhang, L., Feng, C., & Zhang, L. (2015). Vibration cause analysis and elimination of reciprocating compressor inlet pipelines. *Engineering Failure Analysis*, 48(Supplement C), 272-282. <https://doi.org/10.1016/j.engfailanal.2014.11.003>
- Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv:1506.00019 [cs]*. Recuperado de <http://arxiv.org/abs/1506.00019>
- Liu, J. H., & Hu, G. C. (2014). The Study of Vibration for Motor-Compressors. *Applied Mechanics and Materials*, 467, 479-482. <https://doi.org/10.4028/www.scientific.net/AMM.467.479>
- Loja, S., & Vinicio, R. (2017). Diagnóstico de fallos en cajas de engranajes mediante la aplicación de diferentes técnicas de inteligencia artificial. Recuperado de <http://e-spacio.uned.es/fez/view/tesisuned:ED-Pg-TecInd-Rvsanchez>

- Loparo, K. A., Adams, M. L., Lin, W., Abdel-Magied, M. F., & Afshari, N. (2000). Fault detection and diagnosis of rotating machinery. *IEEE Transactions on Industrial Electronics*, 47(5), 1005-1014. <https://doi.org/10.1109/41.873208>
- López, R. F., & Fernández, J. M. F. (2008). *Las Redes Neuronales Artificiales*. Netbiblo.
- Malik, M. A. K. (1979). Reliable Preventive Maintenance Scheduling. *A I I E Transactions*, 11(3), 221-228. <https://doi.org/10.1080/05695557908974463>
- Mandal, P., Senjyu, T., Urasaki, N., & Funabashi, T. (2006). A neural network based several-hour-ahead electric load forecasting using similar days approach. *International Journal of Electrical Power & Energy Systems*, 28(6), 367-373. <https://doi.org/10.1016/j.ijepes.2005.12.007>
- Manepatil, S. S., & Tiwari, A. (2006). Fault diagnosis of reciprocating compressor using pressure pulsations.
- Mayoral, J. V. (2017). *Thomas S. Kuhn: la búsqueda de la estructura*. Prensas de la Universidad de Zaragoza.
- Mendieta, D. R. C. (2018). *Modelado de sistemas dinámicos con MachineLearning: aplicaciones al mantenimiento basado en la condición* (<http://purl.org/dc/dcmitype/Text>). Universidad de Sevilla. Recuperado de <https://dialnet.unirioja.es/servlet/tesis?codigo=140228&info=resumen&idioma=SPA>
- Montilla M., C. A., Arroyave, J. F., & Silva M., C. E. (2007). Caso de aplicación centrado en la confiabilidad RCM, previa existencia de mantenimiento preventivo. *Scientia Et Technica*, XIII(37). Recuperado de <http://www.redalyc.org/resumen.oa?id=84903746>
- Márquez, B. I. P. (2017). *Galileo Galilei*. Lulu.com.
- Na8. (2017, septiembre 11). 7 pasos del Machine Learning para construir tu máquina. Recuperado 4 de octubre de 2018, de <http://www.aprendemachinelearning.com/7-pasos-machine-learning-construir-maquina/>

- Niu, G., Yang, B.-S., & Pecht, M. (2010). Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. *Reliability Engineering & System Safety*, 95(7), 786-796. <https://doi.org/10.1016/j.ress.2010.02.016>
- Oppenheimer, C. H., & Loparo, K. A. (2002). Physically based diagnosis and prognosis of cracked rotor shafts. En *Component and Systems Diagnostics, Prognostics, and Health Management II* (Vol. 4733, pp. 122-133). International Society for Optics and Photonics. <https://doi.org/10.1117/12.475502>
- Ortiz, C. G., & Pérez, I. A. (2016). *Adquisición de señales vibracionales y emisiones acústicas combinando fallos en maquinaria rotativa y elaboración de guías de práctica sobre detección de fallos en engranajes por medio de emisiones acústicas* (Tesis de pregrado). Universidad Politécnica Salesiana, Cuenca, Ecuador.
- Pavía, V. F., & Soto, Á. P. G. (2006). *Evaluación para la mejora de los centros docentes: construcción del conocimiento*. WK Educación.
- Pichler, K., Lughofer, E., Pichler, M., Buchegger, T., Klement, E. P., & Huschenbett, M. (2016). Fault detection in reciprocating compressor valves under varying load conditions. *Mechanical Systems and Signal Processing*, 70-71(Supplement C), 104-119. <https://doi.org/10.1016/j.ymssp.2015.09.005>
- Platón. (1968). *Diálogos de Platón*. Ediciones Ibéricas y L.C.L.
- Popper, K. R. (2008). *La lógica de la investigación científica*. Tecnos.
- Posada C, V. C. (2017). *Propuesta de Diseño y Construcción de un Equipo para Diagnóstico de Fallas en Compresores Herméticos*. Universidad de El Salvador, El Salvador.
- Potočník, P., & Govekar, E. (2016). Vibration-based condition monitoring of compressors by principal component analysis and discriminant analysis. En *The 23rd international conference on sound and vibration, Athens, Greece* (pp. 10–14).

- Ramesh, R. (2007). Compressor valve failure detection and prognostics. *Masters Theses*. Recuperado de http://scholarsmine.mst.edu/masters_theses/5170
- Rao, J. S. (1996). *Rotor Dynamics*. New Age International.
- Ron, J. M. S. (2011). *El jardín de Newton: La ciencia a través de su historia*. Grupo Planeta Spain.
- Saidur, R., Rahim, N. A., & Hasanuzzaman, M. (2010). A review on compressed-air energy use and energy savings. *Renewable and Sustainable Energy Reviews*, 14(4), 1135-1153. <https://doi.org/10.1016/j.rser.2009.11.013>
- Salvador, A. G. (1988). *Introducción a la neumática*. Marcombo.
- Schantz, C. J. (2011). *Non-intrusive fault detection in reciprocating compressors* (Tesis de máster). Massachusetts Institute of Technology.
- Sekhar, A. S. (2004). Model-based identification of two cracks in a rotor system. *Mechanical Systems and Signal Processing*, 18(4), 977-983. [https://doi.org/10.1016/S0888-3270\(03\)00041-4](https://doi.org/10.1016/S0888-3270(03)00041-4)
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & Freitas, N. de. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1), 148-175. <https://doi.org/10.1109/JPROC.2015.2494218>
- Stiaccini, I., Galoppi, G., Ferrari, L., & Ferrara, G. (2016). A reciprocating compressor hybrid model with acoustic FEM characterization. *International Journal of Refrigeration*, 63(Supplement C), 171-183. <https://doi.org/10.1016/j.ijrefrig.2015.10.036>
- Tian, Z., Wu, B., & Chen, M. (2014). Condition-based maintenance optimization considering improving prediction accuracy. *The Journal of the Operational Research Society*; Abingdon, 65(9), 1412-1422. <http://bibliotecavirtual.ups.edu.ec:2099/10.1057/jors.2013.65>
- Tran, V. T., AlThobiani, F., & Ball, A. (2014). An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks. *Expert Systems with Applications*, 41(9), 4113-4122. <https://doi.org/10.1016/j.eswa.2013.12.026>

- Understanding LSTM Networks -- colah's blog. (s. f.). Recuperado 7 de octubre de 2018, de <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Valenzuela, R., & César, J. (2008). Diagnóstico de fallos en sistemas industriales basado en razonamiento borroso y posibilístico. *Riunet*. <https://doi.org/10.4995/Thesis/10251/1922>
- Verma, N. K., Gupta, R., Sevakula, R. K., & Salour, A. (2014). Signal transforms for feature extraction from vibration signal for air compressor monitoring. En *TENCON 2014 - 2014 IEEE Region 10 Conference* (pp. 1-6). <https://doi.org/10.1109/TENCON.2014.7022275>
- Verma, N. K., Sevakula, R. K., Dixit, S., & Salour, A. (2016). Intelligent Condition Based Monitoring Using Acoustic Signals for Air Compressors. *IEEE Transactions on Reliability*, 65(1), 291-309. <https://doi.org/10.1109/TR.2015.2459684>
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2017). Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 652-663. <https://doi.org/10.1109/TPAMI.2016.2587640>
- Virtanen, T., Plumbley, M. D., & Ellis, D. (2017). *Computational Analysis of Sound Scenes and Events*. Springer.
- Volf, B. M. (2017). *A study of reciprocating compressor valve dynamics* (Master). University of West Bohemia Faculty of Mechanical Engineering, Pilsen, Czech Republic.
- Wang, W. (2002). Toward dynamic model-based prognostics for transmission gears. En *Component and Systems Diagnostics, Prognostics, and Health Management II* (Vol. 4733, pp. 157-168). International Society for Optics and Photonics. <https://doi.org/10.1117/12.475505>
- Yang, B.-S., Hwang, W.-W., Kim, D.-J., & Chit Tan, A. (2005). Condition classification of small reciprocating compressor for refrigerators using artificial neural networks and support vector machines. *Mechanical Systems and Signal Processing*, 19(2), 371-390. <https://doi.org/10.1016/j.ymssp.2004.06.002>

- Yang, M. (2009). Air compressor efficiency in a Vietnamese enterprise. *Energy Policy*, 37(6), 2327-2337. <https://doi.org/10.1016/j.enpol.2009.02.019>
- Yin, S., Ding, S. X., Xie, X., & Luo, H. (2014). A Review on Basic Data-Driven Approaches for Industrial Process Monitoring. *IEEE Transactions on Industrial Electronics*, 61(11), 6418-6428. <https://doi.org/10.1109/TIE.2014.2301773>
- Zhao, H., Wang, J., Xing, J., & Gao, Y. (2015). A feature extraction method based on LMD and MSE and its application for fault diagnosis of reciprocating compressor. *Journal of Vibroengineering*, 17(7), 3515-3526.
- Žilka, M. (2014). Methodology for Selecting the Appropriate Maintenance Strategy for Production Machines.

ANEXO A. Códigos de programación para el pre-procesamiento (Autoría: Grupo GIDTEC).

```
def rectified_average(signal):
    """
    Compute rectified average feature
    :param signal: time-series
    :type signal: numpy array
    :return: rectified average of signal
    :rtype: float
    """
    return np.mean(np.abs(signal))

def statistic_features(signal):
    """
    Compute a group of statistical features
    :param signal: time-series
    :type signal: numpy array
    :return: group of statistical feature from the signal
    :rtype: tuple
    """
    mean = np.mean(signal)
    rms = np.sqrt(np.mean(np.square(signal)))
    std_dev = np.std(signal)
    kurtosis = scipy.stats.kurtosis(signal)
    peak = np.max(signal)
    crest = peak / rms
    r_mean = rectified_average(signal)
    form = rms / r_mean
    impulse = peak / r_mean
    variance = std_dev ** 2
    minimum = np.min(signal)

    return mean, rms, std_dev, kurtosis, peak, crest, r_mean, form, impulse, variance, minimum

def build_subsets_labels(labels):
    number = [int(re.split('R|P|.mat', file)[-2]) for file in labels]
    number = np.array(number, dtype=int)
    labels = np.array(labels)
    labels = labels[number <= 17]
    number = number[number <= 17]

    files_train, files_tmp, _, number_tmp = train_test_split(labels, number, train_size=0.7, stratify=
    files_val, files_test, _, _ = train_test_split(files_tmp, number_tmp, train_size=0.5, stratify=
    return files_train, files_val, files_test

class Data_set:
    def __init__(self, path, path_out,
                  sensor_number=5,
                  time_steps=500,
                  increment=250,
                  window=100,
                  name_acq='Analog50k',
                  sample_frequency=50000):

        self.name_acq = name_acq
        self.sensor_number = sensor_number
        self.time_steps = time_steps
        self.increment = increment
        self.window = window
        self.sample_frequency = sample_frequency
        self.processing_dataset(path, path_out)

    def processing_dataset(self, path, path_out):
```

```

files = []
labels = []
for dir in os.listdir(path):
    folder = join(path, dir)
    list_files = [join(folder, file) for file in os.listdir(folder)]
    labels.extend(os.listdir(folder))
    files.extend(list_files)
files_train, files_val, files_test = build_subsets_labels(files)

features = []
labels = []
p = Pool()
iterador = p.imap(self.processing_file, files_train)
#iterador = p.imap(self.cutting_file, files_train)
for iteracion in iterador:
    feature, label = iteracion
    features.extend(feature)
    labels.extend(label)
features = np.array(features).squeeze()
labels = np.array(labels).squeeze()
with open('../data/'+path_out+'/train_set.pickle', 'wb') as f:
    pickle.dump((features, labels), f, pickle.HIGHEST_PROTOCOL)
print('Finish building train set')
p.close()

features = []
labels = []
p = Pool()
iterador = p.imap(self.processing_file, files_val)
#iterador = p.imap(self.cutting_file, files_val)
for iteracion in iterador:
    feature, label = iteracion
    features.extend(feature)
    labels.extend(label)
features = np.array(features).squeeze()
labels = np.array(labels).squeeze()
with open('../data/'+path_out+'/val_set.pickle', 'wb') as f:
    pickle.dump((features, labels), f, pickle.HIGHEST_PROTOCOL)
print('Finish building validation sedft')
p.close()

features = []
labels = []
p = Pool()
iterador = p.imap(self.processing_file, files_test)
#iterador = p.imap(self.cutting_file, files_test)
for iteracion in iterador:
    feature, label = iteracion
    features.extend(feature)
    labels.extend(label)
features = np.array(features).squeeze()
labels = np.array(labels).squeeze()
with open('../data/'+path_out+'/test_set.pickle', 'wb') as f:
    pickle.dump((features, labels), f, pickle.HIGHEST_PROTOCOL)
print('Finish building test set')
p.close()

def processing_dataset_features(self, path, path_out):
    """
    Method to process the dataset in parallel
    :param path: path to raw signals dataset
    :type path: string
    :return: features dataset
    :rtype: numpy array
    """
    files = []
    labels = []

```

```

for dir in os.listdir(path):
    folder = join(path, dir)
    list_files = [join(folder, file) for file in os.listdir(folder)]
    labels.extend(os.listdir(folder))
    files.extend(list_files)
files_train, files_val, files_test = build_subsets_labels(files)

features = []
labels = []
p = Pool()
iterador = p.imap(self.processing_file_features, files_train)
for iteracion in iterador:
    feature, label = iteracion
    if features != []:
        features.extend(feature)
        labels.extend(label)
features = np.array(features).squeeze()
labels = np.array(labels).squeeze()
with open('../data/'+path_out+'/train_set.pickle', 'wb') as f:
    pickle.dump((features, labels), f, pickle.HIGHEST_PROTOCOL)

print('Finish building train set')
p.close()

p = Pool()
iterador = p.imap(self.processing_file_features, files_val)
features = []
labels = []
for iteracion in iterador:
    feature, label = iteracion
    features.extend(feature)
    labels.extend(label)
features = np.array(features).squeeze()
labels = np.array(labels).squeeze()
with open('../data/'+path_out+'/val_set.pickle', 'wb') as f:
    pickle.dump((features, labels), f, pickle.HIGHEST_PROTOCOL)
print('Finish building validation set')
p.close()

p = Pool()
iterador = p.imap(self.processing_file_features, files_test)
features = []
labels = []

for iteracion in iterador:
    feature, label = iteracion
    features.extend(feature)
    labels.extend(label)
features = np.array(features).squeeze()
labels = np.array(labels).squeeze()
with open('../data/'+path_out+'/test_set.pickle', 'wb') as f:
    pickle.dump((features, labels), f, pickle.HIGHEST_PROTOCOL)
print('Finish building test set')
p.close()

def processing_file(self, file):
    """
    Method to process a raw signal file
    :param file: name of raw signal .mat file
    :type file: string
    :return: group of features for the signal in .mat file
    :rtype: numpy array
    """
    label = re.split('R|P|.mat', file)[-3:-1]
    #print(file)
    try:
        data = sio.loadmat(file)
        #signal = data['data'][self.name_acq][0][0][self.sensor_number]
        signal = data['data'][self.name_acq][0][0][0][0][self.sensor_number]

```

```

        ini = 0
        sub_sampled_signal = []
        while ini + self.time_steps <= signal.shape[0]:
            sub_sampled_signal.append(statistic_features(signal[ini:self.time_steps + ini])
            ini += self.increment
        sub_sampled_signal = np.array(sub_sampled_signal)

        ini = 0
        group_signals = []
        group_label = []
        while ini + self.window <= sub_sampled_signal.shape[0]:
            group_signals.append(sub_sampled_signal[ini:self.window + ini])
            group_label.append(label)
            ini += 20
        group_signals = np.array(group_signals)

        return group_signals, group_label
    except:

        print('Falla:', file)
        return [], []

def cutting_file(self, file):
    """
    Method to process a raw signal file
    :param file: name of raw signal .mat file
    :type file: string
    :return: group of features for the signal in .mat file
    :rtype: numpy array
    """
    label = re.split('R|P|.mat', file)[-3:-1]
    #print(file)
    try:
        data = sio.loadmat(file)
        #signal = data['data'][self.name_acq][0][0][self.sensor_number]
        signal = data['data'][self.name_acq][0][0][0][0][self.sensor_number]

        sub_sampled_signal = signal
        group_signals = []
        group_label = []
        ini = 0

        for i in range(50):
            group_signals2 = []
            group_label.append(label)
            for j in range(100):
                group_signals2.append(sub_sampled_signal[ini:self.window + ini])
                ini += self.window
            group_signals.append(np.array(group_signals2).squeeze())
        group_signals = np.array(group_signals)

        return np.moveaxis(group_signals, -1, 0).reshape((50, -1, 100)), group_label
    except:
        print('Falla:', file)
        return [], []

def processing_file_features(self, file):

    label = re.split('R|P|.mat', file)[-3:-1]
    print(file)
    data = sio.loadmat(file)
    signal = data['data'][self.name_acq][0][0][0][0][self.sensor_number]

    features = statistic_features(signal)

    return [features], [label]

def parametric_generation(time_steps, increment, window):
    path = '/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/'

```

```

#path = '/home/titan/databases/PittingGearbox/'
path_out = 'signalsValveCompressor'
#path_out = 'signalsPittingSeverityGearbox'

if not os.path.exists('../data/' + path_out):
    os.makedirs('../data/' + path_out)
# Datasets generation process
dataset = Data_set(path, path_out, time_steps=time_steps, increment=increment, window=window)

if __name__ == "__main__":
    np.random.seed(1234)
    # Path to raw dataset
    path = '/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/'
    path_out = 'datasetValveCompressor'

    if not os.path.exists('../data/' + path_out):
        os.makedirs('../data/' + path_out)
    # Datasets generation process
    dataset = Data_set(path, path_out, time_steps=500, increment=250, window=100)

```


ANEXO B. Códigos de programación para la creación del modelo (Autoría: Grupo GIDTEC).

```

MAX_EVALS = 20

def train_lstm(config, batch_size, dropout):
    tf.reset_default_graph()
    tf.set_random_seed(1234)
    # Instantiate a model
    model = Model(config)

    """Session time"""
    sess = tf.Session() # Depending on your use, do not forget to close the session
    np.random.seed(1234)
    sess.run(model.init_op)

    try:
        acc = 0
        i_best = 0
        np.random.seed(1234)
        for i in range(max_iterations):
            X_batch, y_batch = sample_batch(X_train, y_train, batch_size)
            # X_batch = X_batch.reshape((X_batch.shape[0], X_batch.shape[1], 1))
            # Next line does the actual training
            cost_train, acc_train, _ = sess.run([model.cost, model.accuracy, model.train_op],
                                                feed_dict={model.input: X_batch, model.labels:
                                                            model.labels,
                                                            model.keep_prob: dropout})

            if i % 5 == 1:
                true_labels, pred_labels = sess.run([model.labels, model.prediction],
                                                    feed_dict={model.input: X_batch, model.labels:
                                                                model.labels,
                                                                model.keep_prob: 1.0})

                # Evaluate validation performance
                X_batch, y_batch = X_val, y_val # sample_batch(X_val, y_val, batch_size)
                # X_batch = X_batch.reshape((X_batch.shape[0], X_batch.shape[1], 1))
                cost_val, summ, acc_val = sess.run([model.cost, model.merged, model.accuracy],
                                                  feed_dict={...})

                true_labels, pred_labels = sess.run([model.labels, model.prediction],
                                                    feed_dict={...})

                cm_v = confusion_matrix(true_labels, pred_labels)
                cm_v = cm_v.astype('float') / cm_v.sum(axis=1)[:, np.newaxis]
                print('At %5.0f/%5.0f: COST %5.3f/%5.3f -- Acc %5.3f/%5.3f' % (
                    i, max_iterations, cost_train, cost_val, acc_train, acc_val))

                accura_val = (np.diag(cm_v)).sum() / cm_v.sum()
                print('Exactitud validacion:\n', accura_val)

                if accura_val > acc:
                    acc = accura_val
                    i_best = i
                    true_labels, pred_labels = sess.run([model.labels, model.prediction],
                                                        feed_dict={model.input: X_test, model.labels:
                                                                    model.labels,
                                                                    model.keep_prob: 1.0})

                    cm = confusion_matrix(true_labels, pred_labels)
                    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
                    #print('Test cm:\n', cm)

        sess.close()

        # print(classification_report(true_labels, pred_labels))
    except KeyboardInterrupt:
        pass
    return acc, i_best, cm

```

```

def objective(params):
    # Make sure parameters that need to be integers are integers
    for parameter_name in ['num_layers', 'hidden_size', 'batch_size']:
        params[parameter_name] = int(params[parameter_name])

    batch_size = params['batch_size']
    dropout = 0.8

    config = {'num_layers': params['num_layers'], # number of layers of stacked RNN's
              'hidden_size': params['hidden_size'], # memory cells in a layer
              'max_grad_norm': 5, # maximum gradient norm during training
              'batch_size': batch_size,
              'learning_rate': params['learning_rate'],
              'sl': sl,
              'input_number': n_inputs,
              'num_classes': num_classes}

    epochs = np.floor(batch_size * max_iterations / N)
    print('Train %.0f samples in approximately %d epochs' % (N, epochs))

    start = timer()

    acc, i_best, cm = train_lstm(config, batch_size, dropout)

    run_time = timer() - start
    # Loss must be minimized
    loss = 1 - acc
    # Write to the csv file ('a' means append)
    of_connection = open(out_file, 'a')
    writer = csv.writer(of_connection)
    writer.writerow([loss, params, i_best, run_time, cm.tolist()])

    # Dictionary with information for evaluation
    return {'loss': loss, 'params': params,
            'best_iter': i_best,
            'train_time': run_time, 'status': STATUS_OK}

for time_steps in range(500, 1001, 100):
    for window in range(100, 181, 20):
        for increment in range(int(time_steps / 2), time_steps + 1, int(time_steps / 2)):
            np.random.seed(1234)
            print('*****')
            print('time-steps:', time_steps)
            print('increment:', increment)
            print('window:', window)
            # Set these directories
            direc = '../data'
            summaries_dir = 'summary'
            """Load the data"""
            parametric_generation(time_steps, increment, window)
            X_train, X_val, X_test, y_train, y_val, y_test = load_own_data(direc=direc,
                                                                           dataset="signalsVal")

            N, sl, n_inputs = X_train.shape
            num_classes = len(np.unique(y_train))
            max_iterations = 200

            # Define the search space
            space = {
                'learning_rate': hp.loguniform('learning_rate', np.log(0.001), np.log(0.2)),
                'num_layers': hp.quniform('num_layers', 1, 3, 1),

                'hidden_size': hp.quniform('hidden_size', 2, 20, 2),
                'batch_size': hp.quniform('batch_size', 1000, 10000, 1000)
            }

            # Keep track of results
            bayes_trials = Trials()

```

```

# File to save first results
out_file = '../resultsPitting/S2/lstm_trials'+'_ts'+str(time_steps)+'_a'+str(incre
of_connection = open(out_file, 'w')
writer = csv.writer(of_connection)

# Write the headers to the file
writer.writerow(['loss', 'params', 'best_iter', 'train_time', 'cm'])
of_connection.close()

best = fmin(fn = objective, space = space, algo = tpe.suggest,
            max_evals = MAX_EVALS, trials = bayes_trials, rstate = np.random.Randc

# Sort the trials with lowest loss (highest AUC) first# Sort t
bayes_trials_results = sorted(bayes_trials.results, key = lambda x: x['loss'])

print(bayes_trials_results[:2])

```

ANEXO C. Códigos de programación para evaluación del modelo (Autoría: Grupo GIDTEC).

```

print('Identificacion de los mejores')

tableBest = pd.DataFrame()
for time_steps in range(500,1001,100):
    for window in range(100,181,20):
        for increment in range(int(time_steps / 2), time_steps + 1, int(time_steps / 2)):
            path = '../resultsPitting/S1/lstm_trials' + '_ts' + str(time_steps) + '_a' + str
            results = pd.read_csv(path)
            results.sort_values('loss', ascending=True, inplace=True)
            results.reset_index(inplace=True, drop=True)
            tableBest = tableBest.append(results.loc[0], ignore_index=True)
            tableBest.loc[tableBest.index[-1], 'ts'] = time_steps
            tableBest.loc[tableBest.index[-1], 'win'] = window
            tableBest.loc[tableBest.index[-1], 'inc'] = increment

tableBest.sort_values('loss', ascending=True, inplace=True)
tableBest.reset_index(inplace=True, drop=True)
print(tableBest.head() [['loss', 'ts', 'win', 'inc']])

"""
print('Señales de vibración y espectros')
data = sio.loadmat('/home/titan/databases/RawData/Raw_Data_Valve_Faults_DB_011V0/P1/R1F1L1P1
signal = data['data']['Analog50k'][0][0][0][0][5]
signal = signal[:25000]
time = np.arange(len(signal))/50000
plt.plot(time,signal,linewidth=0.01)
plt.xlabel('time (s)')
plt.ylabel('acceleration (mm/s^2)')
plt.savefig('../results/figVib5.pdf')

T = 1/50000
N = signal.shape[0]
print(N)
yf = scipy.fftpack.fft(signal)
xf = np.linspace(0.0, 1.0/(2.0*T), N/2)
plt.plot(xf, 2.0/N*np.abs(yf[:N/2]),linewidth=0.01)
plt.ylim([0,0.03])
plt.xlabel('frequency (hz)')
plt.ylabel('amplitude')
plt.savefig('../results/figSpec5.pdf')
plt.close()
print('Indicadores de la condición')
direc = '../data'

dataset="signalsValveCompressor"
datadir = direc + '/' + dataset
with open(datadir+'train_set.pickle', "rb") as f:
    data_train, target_train = pickle.load(f)

print(data_train.shape)
plt.plot(data_train[0,:,10],linewidth=0.01,color='r')
plt.savefig('../results/test.pdf')

plt.close()
"""
print('Grafica ts')
matrix_results = np.zeros((6,10))
i = 0
for time_steps in range(500,1001,100):
    list_results = []
    for window in range(100,181,20):

```

```

        for increment in range(int(time_steps / 2), time_steps + 1, int(time_steps / 2)):
            path = '../resultsPitting/S1/lstm_trials' + '_ts' + str(time_steps) + '_a' + str
            results = pd.read_csv(path)
            results.sort_values('loss', ascending=True, inplace=True)

            results.reset_index(inplace=True, drop=True)
            list_results.append(results.loc[0, 'loss'])
        matrix_results[i] = np.array(list_results)
        i += 1
plt.boxplot(matrix_results.transpose())
plt.xlabel('window length')
plt.ylabel('error')
plt.savefig('../resultsPitting/S1/window_length.pdf')
plt.close()

plt.close()
print('Grafica subs length')
matrix_results = np.zeros((5,12))
i = 0
for window in range(100,181,20):
    list_results = []
    for time_steps in range(500,1001,100):
        for increment in range(int(time_steps / 2), time_steps + 1, int(time_steps / 2)):
            path = '../resultsPitting/S1/lstm_trials' + '_ts' + str(time_steps) + '_a' + str
            results = pd.read_csv(path)
            results.sort_values('loss', ascending=True, inplace=True)

            results.reset_index(inplace=True, drop=True)
            list_results.append(results.loc[0, 'loss'])
        matrix_results[i] = np.array(list_results)
        i += 1
plt.boxplot(matrix_results.transpose())
plt.xlabel('subsignal length')
plt.ylabel('error')
plt.savefig('../resultsPitting/S1/subsignal_length.pdf')
plt.close()

print('Comparaciones')
with open('../resultsPitting/S1/model_results.pickle', "rb") as f:
    comparison = pickle.load(f)
    comparison = np.flip(comparison,0)
plt.boxplot(comparison.transpose())
plt.xlabel('model')
plt.ylabel('accuracy')
plt.savefig('../resultsPitting/S1/comparison.pdf')
plt.close()
print('Tabla estadísticos')
for i in range(5):
    col1 = 'LSTM-'+str(i+1)

    col2 = round(np.min(comparison[i]),3)
    col3 = round(np.mean(comparison[i]),3)
    col4 = round(np.max(comparison[i]),3)
    ks = stats.kstest((comparison[i,:]-np.mean(comparison[i,:]))/np.std(comparison[i,:]), 'nor
    col5 = round(ks[1],2)
    col6 = round(np.std(comparison[i]),3)
    print(col1,col5, col3, col6, col2, col4, sep=' & ', end='\\\\\\ \n')

for i in range(5):
    print('LSTM-',i+1,sep='',end=' & ')
    for j in range(i+1,5):
        tt = stats.ttest_ind(comparison[i,:],comparison[j,:],equal_var=False)
        print(round(tt[1],2),end=' & ')
    print('\\\\\\')

"""
time_steps = 48
print('Tabla 1')
for station in range(9):

```



```

ahead = 1
results = pd.read_csv('../results/lstm_trials'+ '_ts'+str(time_steps)+'_a'+str(ahead)+'_s'+
results.sort_values('loss', ascending=True, inplace=True)
results.reset_index(inplace=True, drop=True)
params = ast.literal_eval(results.loc[0, 'params']).copy()
col1 = 'E'+str(station)
col2 = round(results.loc[0, 'loss'],3)
col3 = params['batch_size']
col4 = round(params['learning_rate'],3)
col5 = params['num_layers']
col6 = params['hidden_size']
col7 = results.loc[0, 'best_iter']
col8 = round(results.loc[0, 'train_time'],1)
print(col1,col3,col4,col5,col6,col7,col8,col2,sep=' & ',end='\\\\\\ \n')

print('Tabla 2')
for ahead in range(1,11):
    best = 1
    for station in range(9):
        results_file = pd.read_csv(
            '../results/lstm_trials' + '_ts' + str(time_steps) + '_a' + str(ahead) + '_s' + s
        results_file.sort_values('loss', ascending=True, inplace=True)
        results_file.reset_index(inplace=True, drop=True)
        if results_file.loc[0, 'loss'] < best:
            best = results_file.loc[0, 'loss']
            results = results_file.head()
            s = station

    params = ast.literal_eval(results.loc[0, 'params']).copy()
    col1 = str(ahead)
    col2 = round(results.loc[0, 'loss'], 3)
    col3 = params['batch_size']
    col4 = round(params['learning_rate'], 3)
    col5 = params['num_layers']
    col6 = params['hidden_size']
    col7 = results.loc[0, 'best_iter']
    col8 = round(results.loc[0, 'train_time'], 1)
    col9 = str(s)
    print(col1, col9, col3, col4, col5, col6, col7, col8, col2, sep=' & ', end='\\\\\\ \n')

print('Tabla 3')
for station in range(9):
    ahead = 1
    results = pd.read_csv('../results/lstm_trials'+ '_ts'+str(time_steps)+'_a'+str(ahead)+'_s'+
    results.sort_values('loss', ascending=True, inplace=True)
    results.reset_index(inplace=True, drop=True)

    params = ast.literal_eval(results.loc[0, 'params']).copy()
    col1 = 'E'+str(station)
    vp = round(results.loc[0, 'cm[0][0]', 3)
    fn = round(results.loc[0, 'cm[0][1]', 3)
    fp = round(results.loc[0, 'cm[1][0]', 3)
    vn = round(results.loc[0, 'cm[1][1]', 3)
    prec = round(vp/(vp+fp), 3)
    recall = round(vp/(vp+fn), 3)
    f_val = round(2*prec*recall/(prec+recall), 3)
    acc = round((vp+vn)/(vp+vn+fp+fn), 3)
    print(col1,vp,fn,fp,vn,prec,recall,f_val,acc,sep=' & ',end='\\\\\\ \n')

print('Tabla 4')
for ahead in range(1,11):
    best = 0
    for station in range(9):
        results_file = pd.read_csv(
            '../results/lstm_trials' + '_ts' + str(time_steps) + '_a' + str(ahead) + '_s' + s
        results_file['acc'] = (results_file['cm[0][0]']+results_file['cm[1][1]']) / (results_
        results_file.sort_values('acc', ascending=False, inplace=True)
        results_file.reset_index(inplace=True, drop=True)

```

```

        if results_file.loc[0,'acc'] > best:
            best = results_file.loc[0,'acc']
            results = results_file.head()
            s = station

        params = ast.literal_eval(results.loc[0, 'params']).copy()
        col1 = str(ahead)
        vp = round(results.loc[0, 'cm[0][0]'], 3)
        fn = round(results.loc[0, 'cm[0][1]'], 3)
        fp = round(results.loc[0, 'cm[1][0]'], 3)
        vn = round(results.loc[0, 'cm[1][1]'], 3)
        prec = round(vp / (vp + fp), 3)
        recall = round(vp / (vp + fn), 3)
        f_val = round(2 * prec * recall / (prec + recall), 3)
        acc = round((vp + vn) / (vp + vn + fp + fn), 3)
        col2 = str(s)
        print(col1, col2, vp, fn, fp, vn, prec, recall, f_val, acc, sep=' & ', end='\\n\\n \\n')

print('Grafico 1')
pathFiles = '/home/titan/Dropbox/MaEugenia_Omar/PROGRAMA/'

time_steps = 48
ahead = 10

list_data = []
list_target = []
k = 19
nameFile = join(pathFiles, 'R' + str(k) + '.csv')
df_file = pd.read_csv(nameFile,index_col='Unnamed: 0',parse_dates=True)
df_prec = df_file[['Prec_01','Prec_02',
                    'Prec_03','Prec_04',
                    'Prec_05','Prec_06',
                    'Prec_07','Prec_08','Prec_09']].resample('H').sum()

df_sensor = df_file[df_file.columns.difference(['Prec_01','Prec_02',
                    'Prec_03','Prec_04',
                    'Prec_05','Prec_06',
                    'Prec_07','Prec_08','Prec_09'])].resample('H').mean()
df_per_hour = pd.concat([df_prec,df_sensor],axis=1)
df_per_hour = df_per_hour.loc['2016-04-17':'2016-04-19']
df_per_hour[['Prec_01','Prec_02',
                    'Prec_03','Prec_04',
                    'Prec_05','Prec_06',
                    'Prec_07','Prec_08','Prec_09']].plot(linewidth=0.2)
plt.savefig('../results/im1_a.pdf')

df_per_hour[['Temp_07','Temp_08','Temp_09']].plot(linewidth=0.2)
plt.savefig('../results/im1_b.pdf')

df_per_hour[['VV_07','VV_08','VV_09']].plot(linewidth=0.2)
plt.savefig('../results/im1_c.pdf')

df_per_hour[['Hum_07','Hum_08','Hum_09']].plot(linewidth=0.2)
plt.savefig('../results/im1_d.pdf')
"""

```